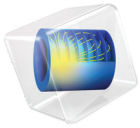


Structural Mechanics Module

User's Guide

6.4

COMSOL

Structural Mechanics Module User's Guide

© 1998–2025 COMSOL

Protected by patents listed on www.comsol.com/patents, or see Help > About COMSOL Multiphysics on the File menu in the COMSOL Desktop for less detailed lists of U.S. Patents that may apply. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement (www.comsol.com/sla) and may be used or copied only under the terms of the license agreement.

COMSOL, the COMSOL logo, COMSOL Multiphysics, COMSOL Desktop, COMSOL Compiler, COMSOL Server, and LiveLink are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see www.comsol.com/trademarks.

Version: COMSOL 6.4

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries or search for an address and phone number. You can also visit the Worldwide Sales Offices page at www.comsol.com/contact/offices for address and contact information.

If you need to contact Support, an online request form is located on the COMSOL Access page at www.comsol.com/support/case. Useful links:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/product-update
- COMSOL Blog: www.comsol.com/blogs
- Discussion Forum: www.comsol.com/forum
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/videos
- Support Knowledge Base: www.comsol.com/support/knowledgebase
- Learning Center: www.comsol.com/support/learning-center

Part number: CM021101

Contents

Chapter 1: Introduction

About the Structural Mechanics Module **38**
A Quick Overview of the Structural Mechanics Module 38
What Problems Can It Solve? 39
The Structural Mechanics Module Physics Interface Guide 42
Common Physics Interface and Feature Settings and Nodes 54
Where Do I Access the Documentation and Application Libraries? 55

Overview of the User's Guide **58**

Chapter 2: Structural Mechanics Modeling

Study Types **67**
Introduction 67
Stationary Analysis 67
Eigenfrequency Analysis 69
Mode Analysis 75
Time-Domain Analysis 76
Frequency-Domain Analysis 77
Mode Superposition 80
Harmonic Perturbation 82
Linearized Buckling Analysis 85
Bolt Pretension Study 86
Random Vibration (PSD) Study 86
Response Spectrum Analysis Study 87
Inertia Relief Study 88

Selecting the Physics Interface **90**
Solid Mechanics 90
Elastic Waves, Time Explicit 94
Shell and Plate 95

Membrane 96
Beam 96
Beam Cross Section. 97
Pipe Mechanics 98
Truss 98
Wire 98

Selecting Discretization **100**
Shape Function Order 100
Lagrange and Serendipity Shape Functions 101
Choosing Shape Functions in Multiphysics Models 101
Implicit Shape Function Orders 102

Using Reduced Integration **104**

Modeling with Plates and Shells **108**
Introduction 108
Dependent Variables 109
Layered and Nonlayered Shells 111
Working with Section Stiffness 113
Loads 113

Modeling with Layered Shells **114**

Modeling with Membrane Elements **120**
About Membranes 120
Membranes for 3D Models 120
Membranes for 2D Axisymmetric Models 120
Stiffness in the Normal Direction. 121
Stabilization 121
Layered and Nonlayered Membranes 122

Modeling with Beam Elements **123**
About Beams 123
In-Plane Beams 123
3D Beam 124
Orientation of the Cross Section. 125
Geometric Variables 126

Result Evaluation 127
Geometric Nonlinearity 129
Modeling Stiffeners 130
Nonrigid Joints 131

Modeling with Truss Elements 135

Modeling Wires and Cables 137

Modeling with the Phase Field in Solids Interface 138
Introduction 138
Dependent Variables 138
Phase–Field Damage Modeling 139

Modeling with the Transport in Solids Interface 140
Introduction 140
Dependent Variables 140
Material and Spatial Variables 141
Mass Transfer Variables 142

Coupling Different Element Types 145
Introduction to the Element Types 145
Coupling Techniques 145

Modeling Embedded Structures and Reinforcements 154

Applying Loads 159
Units, Orientation, and Visualization 159
Load Cases 161
Singular Loads 163
Moments in the Solid Mechanics Interface 165
Pressure 165
Acceleration Loads 165
Temperature Loads — Thermal Expansion 166
Hygroscopic Swelling 166
Total Loads 167
Loads with a Spatial Variation 167
Variables for Loads 168

Defining Constraints **170**
Rigid Body Motion 170
Orientation 172
Prescribed Displacements, Velocities, and Accelerations 172
Symmetry Constraints 174
Elemental and Nodal Constraints 176
Suppressing Constraints on Lower Dimensions 177
Kinematic Constraints 179
Rotational Joints 179
Attachments 179

Calculating Reaction Forces **182**
Using Predefined Variables to Evaluate Reaction Forces 182
Using Weak Constraints to Evaluate Reaction Forces 185
Using Surface Traction to Evaluate Reaction Forces. 186
Evaluating Surface Traction Forces on Interior Boundaries 186

Introduction to Material Models **188**
Material Models for Structural Mechanics 188
Combination of Material Models 189
Entering Material Data 189
Introduction to Linear Elastic Materials 189
Introduction to Viscoelastic Materials 190
Mixed Formulation 190
Modeling Damage 192
About the Material Databases for the Structural Mechanics Module . . 194
Using External Materials 195
References 208

Modeling Piezoelectric Problems **209**
About Piezoelectric Materials 209
Piezoelectric Material Orientation 210
Piezoelectric Losses 218
References for Piezoelectric Damping 225
Piezoelectric Coupling 226
Create the Piezoelectricity Interface and Define Domains 226
Complete Settings of Piezoelectric Materials 228
Add Damping and Loss 229

Define Material Properties 229
Coupling Piezoelectricity with Acoustics 230

Modeling Electrostrictive and Ferroelectroelastic Materials 232
Electrostriction Coupling 232
Ferroelectricity 232
Multiphysics Interfaces 232

Modeling Magnetostrictive Materials 234
Magnetostriction Coupling 234
Linear vs. Nonlinear Magnetostriction 234
Create the Magnetostriction Interface and Define Domains 235

Mechanical Damping and Losses 239
Damping Sources 239
About Damping 240
Rayleigh Damping 242
Loss Factor Damping 244
Viscous Damping 245
Equivalence Between Loss Factor, Rayleigh, and Viscous Damping . . . 245
Explicit Complex-Valued Damping 246
Viscoelastic Damping 246
Piezoelectric Damping 246
Adding Damping in the Modal Solver 247

Modeling Geometric Nonlinearity 248
Geometric Nonlinearity, Frames, and the ALE Method 248
Geometric Nonlinearity for the Solid Mechanics Interface 256
Geometric Nonlinearity for the Shell, Plate, Membrane, Beam, and Truss Interfaces 257
Solving Geometrically Nonlinear Problems 258
Prestressed Structures 258
Geometric Nonlinearity for the Piezoelectric Material 261
References 264

Contact Modeling 265
Overview 265
Setting Up a Contact Problem 266

Contact Pairs 278
General Contact Pairs 281
Meshing for Contact Analysis 282
Settings for Contact Nodes 283
Settings for General Contact Nodes 293
Quasistatic Contact Analysis 296
Implicit Dynamic Contact Analysis 297
Explicit Dynamic Contact Analysis 298
Multiphysics Contact Analysis 300
Special Types of Contact Problems 302
Fallback Conditions to Contact Regions 304
Solver Settings for Contact Analysis. 304
Monitoring the Solution 308
Dependent Variables in Contact Analysis. 309
Important Contact Variables 312
References for Contact Modeling 314

Activating and Deactivating Material **315**

Springs and Dampers **318**

Defining Multiphysics Models **320**
Thermal–Structure Interaction 320
Acoustic–Structure Interaction 321
Thermal–Electric–Structural Interaction 321

Thermally Coupled Problems **323**
Temperature-Dependent Material Data 323
Thermal Expansion 325
Constraints and Thermal Expansion. 331
Thermoelastic Damping 332

Fluid–Structure Interaction **334**
Deforming Fluid Domains 334
Union or Assembly 335
Contact and FSI 336
One-Way FSI 337

Component Mode Synthesis **339**
Introduction 339
COMSOL Implementation 339
Applications 340
Working with CMS Models 341

Computing Mass Properties **353**

Effective Properties of Periodic Structures **356**
Material Homogenization Using the Cell Periodicity Node 356

Modeling Bolts, Fasteners, and Welds **359**
Modeling Pretensioned Bolts 359
Simplified Modeling of Bolt Threads 368
Simplified Modeling of Fasteners 370
Modeling Spot Welds 376
Evaluating Stresses in Continuous Welds 380

Modeling Thin Layers **384**
The Thin Layer Node 384
Interior and Exterior Boundaries 386
Material Models 387
Loads and Boundary Conditions 387
Result Evaluation and Visualization 388

Modeling Cracks **390**
The Crack Node 390
J-Integrals 391
Virtual Crack Extension Method 394
Stress Intensity Factors 395
Contact Between Crack Faces 396

Buckling Analysis **398**
Linearized Buckling Analysis 398
Nonlinear Buckling Analysis 403
Modeling Imperfection in a Buckling Analysis 403

Performing an Explicit Dynamics Analysis **408**
Introduction to Explicit Dynamics 408
Geometry and Meshing for Explicit Dynamics 409
Setting up the Physics 411
Setting Up and Solving an Explicit Dynamics Study 415

Performing a Random Vibration Analysis **421**

Performing a Response Spectrum Analysis **429**

Performing an Inertia Relief Analysis **438**
Introduction 438
Setting up an Inertia Relief Analysis 438

Stress Linearization **440**

Solver Settings for Structural Mechanics **447**
Symmetric Matrices 447
Selecting Iterative Solvers 448
Solver Settings for Viscoelasticity, Creep, and Viscoplasticity 449
Specifying Tolerances and Scaling for the Dependent Variables 450
Solver Settings 450

Result Presentation **452**
Local Orientations 452
How an Expression Is Evaluated 455
Quality Settings 457
Gauss-Point Evaluation 458
Postprocessing of Eigenmodes 460
Plotting Applied Loads 462

Part Libraries **467**

Estimation of Material Parameters **470**

Chapter 3: Structural Mechanics Theory

Solid Mechanics Theory **475**
Introduction 475

Frames and Coordinate Systems **476**
Material and Spatial Coordinates 476
Coordinate Systems. 477
Boundary System. 478

Analysis of Deformation **479**
Lagrangian Formulation 479
Deformation Measures 482
Invariants of Strain 486
Inelastic Strain Contributions 490
Deformed Geometry 494
Axial Symmetry 496
Plane Strain 500
Generalized Plane Strain 500

Stresses **505**
Defining Stress. 505
Invariants of the Stress Tensor 506
Plane Stress 511
Initial Stress and Strain. 512
External Stress. 513
External Strain. 514
Axial Symmetry and Stresses 515

Equation of Motion **517**

Material Models **520**
Linear Elastic Material 523
Mixed Formulation 536
Nonlinear Elastic Materials 538
Linear Viscoelasticity 550
Large Strain Viscoelasticity 582
Hyperelastic Materials 584
Distributed Fiber Models 616
Elastoplastic Materials 621
Porous Plasticity 670
Failure Criteria for Concrete and Rocks 677
Damage Models 683
Coupled Damage–Plasticity 694
Elastoplastic Soils 705
Creep and Viscoplasticity 724
Inelastic Strain Rate 748
Shape Memory Alloy 750
Electromechanics 758
Piezoelectricity 764
Magnetomechanics 773
Magnetostriction and Piezomagnetism 778
Electrostriction 790
Ferroelectroelasticity 792
Rigid Material 795
Safety Factor Evaluation 805

Formulation of the Equilibrium Equations **826**
Equation Implementation 826

Study Types **828**
Frequency-Domain Studies 828
Eigenfrequency Studies 832
Random Vibration Theory 835
Response Spectrum Analysis Theory 842
Linear Buckling 863

Damping **865**
Rayleigh Damping 865
Loss Factor Damping 866
Viscous Damping 867
Maximum Loss Factor 868
Wave Attenuation 869

Loads and Constraints **873**
Distributed Loads 873
Gravity and Base Excitation 874
Frame Acceleration Forces 875
Spring Foundation and Thin Elastic Layer 877
Spring–Damper Theory 881
Added Mass 886
Rigid Motion Suppression 887
Inertia Relief 888
Limited Displacement 890
Low-Reflecting Boundary Condition 891
Cyclic Symmetry and Floquet Periodic Conditions 892
Symmetry Condition with Translation 894
Roller Boundary Condition 895
Thermal Expansion of Constraints 897
Fluid–Structure Interaction 899
Cell Periodicity 900
Rigid Connector 905
Continuity Condition 908

Contact Analysis Theory **912**
Contact Search and Kinematics 913
Kinematics for Interior Boundaries 917
Definition of Contact Tractions 918
Normal Contact 918
Tangential Contact with Friction 925
Unbiased Formulation and Self Contact 931
General Contact Equations 931
Adhesion 932
Decohesion 934
Wear 946

Bolt Modeling Theory **949**
Tightening Torque 949

Theory for Spot Welds **952**
Introduction 952
Stiffness 952
Force Evaluation 953

Theory for Fasteners **955**
Introduction 955
Stiffness 955
Automatic Hole Detection 960
Force Evaluation 961
Safety Factors 962

Stress Linearization **963**

J-Integral Theory **966**

Virtual Crack Extension Theory **970**

Embedded Elements **973**

Thin Layers **977**
Deformation Gradient. 977
Solid Approximation 978
Membrane Approximation 979
Spring Approximation 980
Material Models in Thin Layers 980
Spring Material. 981

Enclosed Fluids **984**
Volume Definition 985
Symmetry Planes 986
Equation of State 987
Fluid Load 988

Reduced Integration and Hourglass Stabilization **990**

Average Rotation **992**

Warpage **994**
Finding the Average Rotation 994
Definition of Warping 997

Energy Quantities **999**
Elastic Energy 999
Kinetic Energy 1001
Dissipated Energy 1001
Mechanical Energy Flux 1002
Energy Variables 1003

Theory for the Shell and Plate Interfaces **1007**
General Theory for the Shell and Plate Interfaces 1007
Theory for FSDT Laminated Shell 1033
Theory for Hyperelastic Shells 1049
Results Evaluation in a Laminate 1049
Theory for Section Stiffness 1050

Theory for the Layered Shell Interface **1052**
About Layered Shells 1052
Governing Equations 1052

Theory for the Membrane Interface **1061**
Introduction 1061
Local Coordinate Systems 1061
Strain–Displacement Relation 1062
Constitutive Relation and Weak Contributions 1063
Wrinkling 1064
External Loads 1068
Stress Calculations 1068

Theory for the Beam Interface **1069**
Shape Functions 1069
Geometric Nonlinearity 1071

Strain–Displacement/Rotation Relation 1074
Stress–Strain Relation 1074
Thermal Strain 1075
Hygroscopic Swelling 1075
Initial Stress and Strain 1076
Implementation 1077
Theory for Section Stiffness 1079
Stress Evaluation 1081
Common Cross Sections 1083

Theory for the Beam Cross Section Interface 1100
Cross-Section Properties 1100
Computation of Stresses 1114

Theory for the Truss Interface 1117
General Theory for the Truss Interface 1117

Theory for the Wire Interface 1126
General Theory for the Wire Interface 1126

Theory for the Pipe Mechanics Interface 1129
About Pipes 1129
Effects of Internal Pressure 1130
Fluid Flow Forces 1134
Effects of Thermal Expansion 1136
Stress Evaluation 1138
Connection Between Pipes and Structures 1143
Cross Sections 1145

Theory for the Phase Field in Solids Interface 1150

Theory for the Transport in Solids Interface 1153
The Transport Equation 1153
Diffusion 1153
Mobility 1154
Mass Flux 1155
Normal Flux 1155
Mass Balance 1156

External Flux 1156
Chemical Potential 1157
Electromigration 1158
Stress Migration 1159
Thermophoresis 1159
Electrode Surface Coupling 1160

References **1161**

Chapter 4: Solid Mechanics

The Solid Mechanics Interface **1176**
Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics . . 1189
Initial Values 1196
Change Thickness 1197
Change Cross Section 1198
Linear Elastic Material 1198
Nonlinear Elastic Material 1209
Elastoplastic Soil Material 1218
Hyperelastic Material 1231
Shape Memory Alloy 1246
Phase Transformation Direction 1254
Piezoelectric Material 1255
Piezomagnetic Material 1262
Magnetostrictive Material 1270
Viscoelasticity 1270
Mullins Effect 1280
Plasticity 1281
Set Variables 1289
Annealing 1291
Pressure-Dependent Plasticity 1291
Creep 1298
Additional Creep 1303
Viscoplasticity 1305
Polymer Viscoplasticity 1313
Porous Plasticity 1319

Soil Plasticity 1327
Cap and Cutoff 1333
Concrete 1335
Rocks 1341
Fiber 1343
Thermal Expansion (Fiber) 1346
Thermal Expansion (for Materials) 1348
Hygroscopic Swelling 1352
Intercalation Strain 1354
Initial Stress and Strain 1355
External Stress 1357
External Strain 1360
Inelastic Strain Rate 1364
Damage 1367
Activation 1372
Safety 1374
Damping 1380
Mechanical Damping 1386
Coupling Loss 1388
Dielectric Loss 1389
Conduction Loss (Time-Harmonic) 1391
External Stress–Strain Relation 1392
Rigid Material 1398
Initial Values (Rigid Material) 1402
Fixed Constraint (Rigid Material) 1403
Prescribed Displacement/Rotation 1404
Applied Force (Rigid Material) 1407
Location Nodes 1409
Applied Moment (Rigid Material) 1410
Mass and Moment of Inertia (Rigid Material) 1411
Center of Mass Nodes 1413
Spring Foundation (Rigid Material) 1414
Free 1416
Prescribed Displacement 1417
Prescribed Velocity 1424
Prescribed Acceleration 1426
Fixed Constraint 1429
Thermal Expansion (for Constraints) 1432

Roller 1434
Symmetry 1437
Symmetry Plane 1440
Antisymmetry 1442
Rigid Motion Suppression 1444
Body Load 1446
Gravity 1449
Base Excitation 1451
Inertia Relief 1452
Rotating Frame 1454
Linearly Accelerated Frame 1458
Boundary Load 1460
Edge Load 1465
Point Load 1466
Point Load, Free 1470
Ring Load 1473
Ring Load, Free 1474
Point Load (on Axis) 1476
Thin Layer 1478
Fiber (Thin Layer) 1480
Prescribed Displacement (Thin Layer) 1483
Fixed Constraint (Thin Layer) 1486
Roller (Thin Layer) 1487
Face Load (Thin Layer) 1488
Boundary Load (Thin Layer) 1492
Spring Material. 1496
Damping (Spring Material) 1499
Spring Foundation 1500
Thin Elastic Layer. 1507
Predeformation 1510
Spring–Damper 1512
Source Point (for Spring–Damper) 1517
Destination Point (for Spring–Damper) 1518
Source Point Nodes. 1518
Destination Point Nodes 1518
Point Mass 1519
Added Mass. 1520
Periodic Condition 1523

Adiabatic Heating 1526
Cell Periodicity 1528
Boundary Pair 1534
Low-Reflecting Boundary 1534
Fasteners 1535
Safety (for Fasteners) 1539
Hole Selection 1540
Rigid Connector 1543
Center of Rotation Nodes 1553
Thermal Expansion (Rigid Connector) 1554
Applied Force (Rigid Connector) 1556
Applied Moment (Rigid Connector) 1558
Mass and Moment of Inertia (Rigid Connector) 1559
Spring Foundation (Rigid Connector) 1561
Attachment 1563
Thermal Expansion (Attachment) 1565
Reduced Flexible Components 1567
Component Definition 1568
Fixed Joint 1569
Source Filter 1571
Destination Filter 1571
Contact 1572
Interior Contact 1585
Friction 1592
Slip Velocity 1597
Adhesion 1599
Decohesion 1602
Wear 1605
Stabilization 1608
General Contact 1610
Contact Model 1613
No Contact 1615
Offset 1616
Damping (General Contact) 1618
Friction (General Contact) 1619
Continuity 1621
Bolt Pretension 1623
Bolt Selection 1626

Bolt Thread Contact 1629
Crack 1631
Crack Closure 1633
Face Load (Crack) 1634
J-Integral 1635
Reverse Crack Front 1638
Virtual Crack Extension 1638
Enclosed Cavity 1640
Fluid (Enclosed Cavity) 1643
Prescribed Pressure (Enclosed Cavity) 1645
Filter (Enclosed Cavity) 1646
Port 1648
Elastic Predeformation 1652
Phase 1654
Harmonic Perturbation 1656
Stress Linearization 1658
Section Forces 1660
Test Material 1663
Local System Results 1666
Average Rotation 1667
Warpage 1670
Wave Speeds 1672
Important Variables in the Solid Mechanics Interface 1674

Chapter 5: Solid Mechanics, Explicit Dynamics

The Solid Mechanics, Explicit Dynamics Interface **1680**
Domain, Boundary, Edge, Point, Global, and Pair Nodes for Solid Mechanics, Explicit Dynamics 1685
Prescribed Displacement (on Domains) 1689
Mass Scaling 1690
Artificial Viscosity 1690

Chapter 6: Shell and Plate

The Shell and Plate Interfaces **1694**
Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces . . . 1699
Initial Values . . . 1703
Thickness and Offset . . . 1703
Thickness Change . . . 1705
Linear Elastic Material . . . 1706
Linear Elastic Material, Layered . . . 1709
Shell Local System . . . 1715
Hyperelastic Material, Layered . . . 1716
Piezoelectric Material, Layered . . . 1719
Viscoelasticity . . . 1724
Plasticity . . . 1725
Set Variables . . . 1727
Creep . . . 1727
Additional Creep . . . 1729
Viscoplasticity . . . 1730
Polymer Viscoplasticity . . . 1732
Additional Network . . . 1733
Fiber . . . 1734
Thermal Expansion (Fiber) . . . 1736
Thermal Expansion (for Materials) . . . 1738
Hygroscopic Swelling . . . 1742
Initial Stress and Strain . . . 1745
External Stress . . . 1748
External Strain . . . 1751
Inelastic Strain Rate . . . 1752
Damage . . . 1753
Mullins Effect . . . 1754
Safety . . . 1755
Damping . . . 1756
Mechanical Damping . . . 1758
Coupling Loss . . . 1759
Dielectric Loss . . . 1760
Section Stiffness . . . 1761

Fixed Constraint 1764
Prescribed Displacement/Rotation 1766
Thermal Expansion (for Constraints) 1771
Prescribed Velocity 1772
Prescribed Acceleration 1775
Pinned 1777
No Rotation 1779
Simply Supported. 1780
Symmetry 1781
Symmetry Plane 1784
Antisymmetry 1786
Body Load 1787
Face Load 1789
Edge Load 1793
Point Load 1799
Point Load, Free 1803
Ring Load 1805
Ring Load, Free 1807
Spring Foundation 1809
Predeformation 1813
Added Mass. 1814
Point Mass 1816
Periodic Condition 1817
Adiabatic Heating, Layered 1821
Boundary to Boundary. 1824
Edge to Boundary 1827
Edge to Edge 1829
Spot Welds 1832
Fasteners. 1836
Safety (for Fasteners) 1841
Hole Selection. 1842
Rigid Connector 1844
Thermal Expansion (Rigid Connector) 1852
Attachment 1853
Thermal Expansion (Attachment). 1854
Phase 1855
Harmonic Perturbation 1856

Chapter 7: Layered Shell

The Layered Shell Interface **1860**
The Shell Properties and Interface Selection Sections 1864
Domain, Boundary, Edge, Point, and Pair Nodes for the Layered Shell Interface 1865
Initial Values 1867
Linear Elastic Material 1868
Hyperelastic Material 1870
Piezoelectric Material 1872
Viscoelasticity 1877
Mullins Effect 1878
Plasticity 1879
Set Variables 1881
Creep 1882
Additional Creep 1883
Viscoplasticity 1884
Polymer Viscoplasticity 1886
Additional Network 1887
Thermal Expansion (for Materials) 1888
Hygroscopic Swelling 1889
Initial Stress and Strain 1891
External Stress 1892
External Strain 1894
Inelastic Strain Rate 1896
Damage 1897
Activation 1899
Delamination 1900
Safety 1903
Damping 1904
Mechanical Damping 1906
Coupling Loss 1907
Dielectric Loss 1908
Rigid Material 1909
Free 1910
Prescribed Displacement 1911
Prescribed Displacement, Interface 1912

Prescribed Velocity 1913
Prescribed Velocity, Interface 1915
Prescribed Acceleration 1916
Prescribed Acceleration, Interface 1917
Fixed Constraint 1918
Fixed Constraint, Interface 1919
Thermal Expansion (for Constraints) 1920
Roller 1922
Roller, Interface 1923
Symmetry 1923
Antisymmetry 1924
Rigid Motion Suppression 1925
Body Load 1926
Face Load 1928
Rotating Frame 1929
Linearly Accelerated Frame 1930
Boundary Load 1931
Edge Load 1933
Line Load. 1934
Point Load 1935
Phase 1936
Spring Foundation 1937
Spring Foundation, Interface 1938
Thin Elastic Layer. 1940
Thin Elastic Layer, Interface 1943
Predeformation 1944
Added Mass. 1946
Added Mass, Interface 1947
Adiabatic Heating. 1948
Rigid Connector 1950
Rigid Connector, Interface 1951
Thermal Expansion (Rigid Connector) 1952
Attachment 1953
Thermal Expansion (Attachment). 1954
Continuity 1955

Chapter 8: Membrane

The Membrane Interface **1960**
Boundary, Edge, Point, and Pair Nodes for the Membrane Interface . . . 1962
Linear Elastic Material . . . 1964
Linear Elastic Material, Layered . . . 1970
Viscoelasticity . . . 1976
Plasticity . . . 1977
Set Variables . . . 1978
Creep . . . 1979
Additional Creep . . . 1980
Viscoplasticity . . . 1982
Wrinkling . . . 1984
Fiber . . . 1985
Thermal Expansion (for Materials) . . . 1987
Hygroscopic Swelling . . . 1989
Initial Stress and Strain . . . 1991
External Stress . . . 1993
Inelastic Strain Rate . . . 1995
Safety . . . 1996
Damping . . . 1998
Symmetry . . . 2000
Antisymmetry . . . 2002
Stabilization . . . 2003
Adiabatic Heating, Layered . . . 2005
Face Load . . . 2007
Edge Load . . . 2009
Ring Load . . . 2011
Ring Load, Free . . . 2013

Chapter 9: Beam

The Beam Interface **2016**
Boundary, Edge, Point, and Pair Nodes for the Beam Interface . . . 2020
Initial Values . . . 2022

Cross-Section Data 2023
Section Orientation 2028
Linear Elastic Material 2030
Thermal Expansion (for Materials) 2032
Hygroscopic Swelling 2033
Initial Stress and Strain. 2035
External Stress. 2036
Section Stiffness 2037
Prescribed Displacement/Rotation 2040
Prescribed Velocity 2043
Prescribed Acceleration 2045
Pinned 2047
Thermal Expansion (for Constraints) 2048
No Rotation 2050
Symmetry 2050
Antisymmetry 2053
Edge Load 2054
Point Load 2058
Point Load, Free 2060
Point Mass 2061
Rigid Connector 2062
Attachment 2069
Beam End Release 2070
Edge Group 2071
Phase 2072
Harmonic Perturbation 2073

Chapter 10: Beam Cross Section

Using the Beam Cross Section Interface **2076**
About Beams and Cross-Section Data 2076
Using the Beam Cross Section Interface 2077

The Beam Cross Section Interface **2084**
Homogeneous Cross Section 2085
Hole 2087

Safety 2089

Chapter 11: Truss

The Truss Interface **2092**
Boundary, Edge, Point, and Pair Nodes for the Truss Interface, Explicit Dynamics 2094
Cross-Section Data 2096
Straight Edge Constraint 2098
Linear Elastic Material 2100
Set Variables 2103
Thermal Expansion (for Linear Elastic Material) 2104
Thermal Expansion (for Shape Memory Alloy) 2105
Hygroscopic Swelling 2107
Initial Stress and Strain. 2107
External Stress. 2108
Spring-Damper Material 2110
Pinned 2111
Thermal Expansion (for Constraints) 2112
Symmetry 2113
Antisymmetry 2115
Edge Load 2117
Point Mass 2119

Chapter 12: Truss, Explicit Dynamics

The Truss, Explicit Dynamics Interface **2122**
Boundary, Edge, Point, Global and Pair Nodes for the Truss, Explicit Dynamics Interface. 2124

Chapter 13: Wire

The Wire Interface **2128**
Boundary, Edge, Point, and Pair Nodes for the Wire Interface 2129
Elastic Wire 2131
Thermal Expansion (for Elastic Wire) 2132
Hygroscopic Swelling 2133
Initial Stress and Strain 2134
Pinned 2134
Thermal Expansion (for Constraints) 2135
Symmetry 2137
Antisymmetry 2139
Edge Load 2140

Chapter 14: Pipe Mechanics

The Pipe Mechanics Interface **2144**
Boundary, Edge, Point, and Pair Nodes for the Pipe Mechanics Interface 2147
Pipe Cross Section 2148
Section Orientation 2152
Bend 2155
Fluid and Pipe Materials 2157
Rigid Material 2159
Thermal Expansion (for Materials) 2161
Gravity 2163
Base Excitation 2164
Linearly Accelerated Frame 2165
Fluid Load 2167

Chapter 15: Phase Field in Solids

The Phase Field in Solids Interface **2170**
Domain and Boundary Nodes for the Phase Field in Solids Interface . . 2171

Phase Field Model 2172
Bounds 2173
Prescribed Phase Field 2174
Source 2175
Initial Values 2175
No Flux 2176
Symmetry 2176

Chapter 16: Transport in Solids

The Transport in Solids Interface 2178
Domain and Boundary Nodes for the Transport in Solids Interface . . . 2180
Chemical Potential 2181
External Chemical Potential 2183
External Flux 2184
Electromigration 2184
Stress Migration 2185
Thermophoresis 2186
Initial Values 2187
No Flux 2187
Prescribed Chemical Potential 2188
Prescribed Transported Quantity 2189
Periodic Condition 2190
Solid 2191
Flux 2193
Source 2194
Symmetry 2194
Electrode Surface Coupling 2195

Chapter 17: Multiphysics Interfaces and Couplings

The Thermal Stress, Solid Interface **2202**

The Thermal Stress, Shell Interface **2204**

The Thermal Stress, Layered Shell Interface **2206**

The Thermal Stress, Membrane Interface **2208**

The Joule Heating and Thermal Expansion Interface **2210**

Overview of Fluid–Structure Interaction Interfaces **2212**

The Fluid–Solid Interaction Interface **2214**

The Fluid–Solid Interaction, Fixed Geometry Interface **2216**

The Fluid–Shell Interaction Interface **2218**

The Fluid–Shell Interaction, Fixed Geometry Interface **2220**

The Fluid–Membrane Interaction Interface **2222**

The Fluid–Membrane Interaction, Fixed Geometry Interface **2224**

The Fluid–Pipe Interaction, Fixed Geometry Interface **2226**

The Solid–Thin-Film Damping Interface **2227**

The Shell–Thin-Film Damping Interface **2229**

The Fluid–Solid Interaction, Viscoelastic Flow Interface **2231**

The Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry

Interface **2233**

The Fluid–Solid Interaction, Conjugate Heat Transfer Interface **2235**

The Fluid–Solid Interaction, Two-Phase Flow, Phase Field Interface **2237**

The Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry Interface **2239**

Overview of Poroelasticity Interfaces **2241**

The Poroelasticity, Solid Interface **2242**
The Poroelasticity, Solid Interface 2242
Poroelasticity (Multiphysics Coupling) 2244

The Poroelasticity, Layered Shell Interface **2247**
Poroelasticity, Layered (Multiphysics Coupling) 2248

The Unsaturated Poroelasticity Interface **2251**
The Unsaturated Poroelasticity Interface 2251
Unsaturated Poroelasticity (Multiphysics Coupling) 2252

Overview of Electromagnetics–Structure Interaction Interfaces **2255**

The Electromechanics, Solid Interface **2257**
Deforming Domain 2257

The Electromechanics, Shell Interface **2259**
Deforming Domain 2259

The Electromechanics, Membrane Interface **2261**
Deforming Domain 2261

The Electromechanics, Boundary Elements Interface **2263**

The Piezoelectricity, Solid Interface **2264**
Coupling Piezoelectricity with Acoustics 2266

The Piezoelectricity, Layered Shell Interface **2268**

The Electrostriction Interface **2271**

The Ferroelectroelasticity Interface **2273**

The Magnetomechanics, Solid Interface **2275**
Deforming Domain 2275

The Magnetomechanics, Shell Interface **2277**
Deforming Domain 2277

The Magnetomechanics, Membrane Interface **2279**
Deforming Domain 2279

The Magnetomechanics, No Currents, Solid Interface **2281**
Deforming Domain 2281

The Magnetomechanics, No Currents, Shell Interface **2283**
Deforming Domain 2283

The Magnetomechanics, No Currents, Membrane Interface **2285**
Deforming Domain 2285

The Magnetic–Elastic Interaction in Rotating Machinery

Interface **2287**

The Piezomagnetism Interface **2289**

The Nonlinear Magnetostriction Interface **2292**

The Phase-Field Damage Interface **2295**

The Phase-Field Damage, Explicit Dynamics Interface **2297**

Multiphysics Couplings **2299**
Thermal Expansion 2299
Thermal Expansion, Thin Layer 2302
Thermal Expansion, Layered 2305
Hygroscopic Swelling 2308
Shrinkage and Swelling 2310
Piezoelectricity 2312
Piezoelectricity, Layered 2313
Electrostriction 2315
Electromechanics, Solid 2318
Electromechanics, Boundary 2320
Electromechanics, Boundary Elements 2322
Magnetomechanics, Solid 2324
Magnetomechanics, Boundary 2328
Magnetic Forces, Rotating Machinery 2331
Piezomagnetism 2334
Nonlinear Magnetostriction 2336
Magnetostriction 2339
Fluid–Structure Interaction 2339
Fluid–Structure Interaction, Pair 2341
Fluid–Pipe Interaction 2343
Structure–Thin-Film Flow Interaction 2345
Shell–Thin-Film Flow Interaction 2346
Phase-Field Damage (Multiphysics Coupling) 2348

Couplings Between Structural Mechanics Interfaces **2353**
Solid–Thin Structure Connection 2353
Solid–Beam Connection 2357

Shell–Beam Connection 2362
Layered Shell–Structure Transition 2364
Layered Shell–Structure Cladding 2366
Layered Shell–Shell Connection 2368
Structure–Pipe Connection 2369
Embedded Reinforcement 2370
Beam Cross Section–Beam Coupling 2374
Beam–Beam Cross Section Coupling 2375

Chapter 18: Glossary

Glossary of Terms **2378**

Index **2385**

1

Introduction

This guide describes the Structural Mechanics Module, an optional add-on package that extends the COMSOL Multiphysics® modeling environment with customized physics interfaces that solve problems in the fields of structural and solid mechanics, including special physics interface for modeling of shells, membranes, beams, plates, trusses, wires, and pipes.

This chapter introduces you to the capabilities of this module and includes a summary of the physics interfaces as well as information about where you can find additional documentation and model examples. The last section is a brief overview with links to each chapter in this guide.

- About the Structural Mechanics Module
- Overview of the User's Guide

About the Structural Mechanics Module

In this section:

- A Quick Overview of the Structural Mechanics Module
- What Problems Can It Solve?
- The Structural Mechanics Module Physics Interface Guide
- Common Physics Interface and Feature Settings and Nodes
- Where Do I Access the Documentation and Application Libraries?

The Physics Interfaces and Building a COMSOL Multiphysics Model in the *COMSOL Multiphysics Reference Manual*

A Quick Overview of the Structural Mechanics Module

The Structural Mechanics Module solves problems in the fields of structural and solid mechanics, adding special physics interfaces for modeling shells and beams, for example.

The physics interfaces in this module are fully multiphysics enabled, making it possible to couple them to any other physics interfaces in COMSOL Multiphysics or the other modules. Available physics interfaces include:

- Solid mechanics for 1D and 2D plane stress, plane strain, and generalized plane strain, 2D axial symmetry, and 3D solids
- Beams in 2D and 3D, Euler and Timoshenko theory
- Pipes in 2D and 3D
- Truss elements in 2D and 3D
- Wires in 2D and 3D
- Shells and plates, Mindlin theory, 3D and 2D axial symmetry
- Membranes, 3D and 2D axial symmetry
- With the Composite Materials Module, the Layered Shell interface is also available.

The module's study capabilities include static, eigenfrequency, time dependent (transient), frequency response, buckling, response spectrum, random vibration, and parametric studies.

There are several material models:

- Linear Elastic Materials can be isotropic, orthotropic, or fully anisotropic, and you can use local coordinate systems to specify material properties.
- Linear Viscoelastic Materials
- Piezoelectric Materials
- Magnetostrictive Materials are available when used together with the AC/DC module.
- Material models for hyperelasticity, metal plasticity, porous plasticity, creep, viscoplasticity, nonlinear elasticity, soil plasticity, concrete, rocks, and clay are available with the optional Nonlinear Structural Materials Module and Geomechanics Module.

Large deformations, as well as contact and friction, can also be modeled.

Coupling structural analysis with thermal analysis is one example of multiphysics easily implemented with the module, which provides predefined multiphysics couplings for thermal stress and other types of multiphysics. Piezoelectric materials, coupling the electric field and strain in both directions are fully supported inside the module through special multiphysics couplings solving for both the electric potential and the displacements. Structural mechanics couplings are common in simulations done with COMSOL Multiphysics and occur in interaction with, for example, fluid flow (fluid–structure interaction, FSI), chemical reactions, acoustics, electric fields, magnetic fields, and optical wave propagation.

What Problems Can It Solve?

The Structural Mechanics Module contains a set of physics interfaces adapted to a broad category of structural-mechanics analyses. The module serves as an excellent tool for the professional engineer, researcher, and teacher. In education, the benefit of the short learning curve is especially useful because educators need not spend excessive time learning the software and can instead focus on the physics and the modeling process.

A short summary is given below.

STATIC ANALYSIS

In a *static analysis*, the load and constraints are fixed in time.

EIGENFREQUENCY ANALYSIS

An *eigenfrequency analysis* finds the damped or undamped eigenfrequencies and mode shapes of a structure, sometimes referred to as the *free vibration* of a structure. Prestress effects and damping can also be taken into account.

TRANSIENT ANALYSIS

A *transient analysis* finds the transient response for a time-dependent model, taking into account mass and mass moment of inertia. The transient analysis can be either direct or using a modal solution.

FREQUENCY RESPONSE ANALYSIS

A *frequency response* analysis finds the steady-state response to harmonic loads. The frequency-response analysis can be either direct or using a modal solution. Effects of prestress can be included.

LINEAR BUCKLING STUDY

A *linear buckling analysis* uses the stiffness coming from stresses and material to define an eigenvalue problem where the eigenvalue is a load factor that, when multiplied with the actual load, gives the critical load in a linear context.

PARAMETRIC ANALYSIS

A *parametric* analysis finds the solution dependence due to the variation of a specific parameter, which could be, for instance, a material property or the position of a load.

RESPONSE SPECTRUM ANALYSIS

Response spectrum analysis provides a method to estimate peak values of, for example, displacements and stresses when a structure is subjected to a short, nondeterministic event like an earthquake or a shock.

RANDOM VIBRATION ANALYSIS

For steady-state random dynamic loading, like wind or waves, it is possible to perform *random vibration* analysis where the input is given in terms of a power spectrum density (PSD).

THERMAL STRESS

In a transient *thermal stress* study, the program neglects mass effects, assuming that the time scale in the structural mechanics problem is much smaller than the time scale in the thermal problem.

LARGE DEFORMATIONS

You can also enable geometric nonlinearity for all structural mechanics interfaces. The engineering strain is then replaced with the Green–Lagrange strain and the stress with the second Piola–Kirchhoff stress. To solve the problem, the program uses a *total Lagrangian formulation*.

ELASTOPLASTIC MATERIALS

An elastoplastic analysis involves a nonlinear material with or without hardening. Several isotropic and kinematic hardening models are available.

The material models allow large strains.

The elastoplastic material models are available in the Solid Mechanics, Shell, Layered Shell, Membrane, and Truss interfaces.

CREEP AND VISCOPLASTIC MATERIALS

A number of different material models for creep and viscoplasticity are available. In these materials, the rate of strain depends on the stress.

HYPERELASTIC MATERIALS

In hyperelastic materials, the stresses are computed from a strain energy density function. They are often used to model rubber and biological tissue, but are also used in acoustic elasticity. Many different models are available.

The hyperelastic materials are available in the Solid Mechanics, Shell, Layered Shell, and Membrane interfaces.

NONLINEAR ELASTIC MATERIALS

The nonlinear elastic materials are intended for materials that exhibit a nonlinear behavior already at small strains. Some brittle material as well as soils show this behavior.

VISCOELASTIC MATERIALS

Viscoelastic materials have a time-dependent response, even if the loading is constant. Viscoelasticity is available in the Solid Mechanics, Shell, Layered Shell, and Membrane interfaces.

- Hyperelastic, elastoplastic, creep, viscoplastic, and nonlinear elastic material models are available with the Nonlinear Structural Materials Module.
- Additional functionality and material models for geomechanics and soil mechanics — nonlinear elasticity, soil plasticity, concrete, rock, and clay material models — is available with the Geomechanics Module.

CONTACT MODELING

You can model contact between parts of a structure. The Solid Mechanics, Shell, Layered Shell, and Membrane interfaces support contact with or without friction. Three contact algorithms are available: *penalty*, *augmented Lagrangian*, and *Nitsche* methods. The contact models can be augmented with adhesion and decohesion.

SPECIAL MODELING TECHNIQUES

In the Structural Mechanics Module, you will find support for many important special modeling techniques. Some examples are:

- Bolt pretension
- Bolt thread modeling
- Stress linearization
- Rigid connectors
- Transitions between solid, shell, and beam elements

The Structural Mechanics Module Physics Interface Guide

The Poroelasticity interface requires and is coupled with the Structural Mechanics Module, and is discussed in the *Subsurface Flow Module User's Guide*.

At any time, a new model can be created or physics interfaces added. Right-click the **Root** (top) node and select **Add Component** or right-click a **Component** node and select **Add Physics**.

Depending on the physics interface, specify parameters defining a problem on points, edges (3D), boundaries, and domains. It is possible to specify loads and constraints on all available geometry levels, but material properties can only be specified for the domains, except for shells, membranes, beams, and trusses, where they are defined on the boundary or edge level.

In the *COMSOL Multiphysics Reference Manual:*

- Studies and Solvers
- The Physics Interfaces
- For a list of all the core physics interfaces included with a COMSOL Multiphysics license, see Physics Interface Guide.

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Acoustics				
Elastic Waves				
Elastic Waves, Time Explicit		elte	3D, 2D, 2D axisymmetric	time dependent
Piezoelectric Waves, Time Explicit		—	3D, 2D, 2D axisymmetric	time dependent
Chemical Species Transport				
Transport in Solids		ts	all dimensions	stationary; time dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Fluid Flow				
Fluid–Structure Interaction				
Fluid–Solid Interaction[2]		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Fluid–Shell Interaction[2]		—	3D, 2D axisymmetric	stationary; time dependent
Fluid–Shell Interaction, Fixed Geometry		—	3D, 2D axisymmetric	stationary; stationary, one way; time dependent; time dependent, one way
Fluid–Membrane Interaction[2]		—	3D, 2D axisymmetric	stationary; time dependent
Fluid–Membrane Interaction, Fixed Geometry[2]		—	3D, 2D axisymmetric	stationary; stationary, one way; time dependent; time dependent, one way
Fluid–Pipe Interaction, Fixed Geometry[2,7]		—	3D, 2D	stationary; time dependent
Conjugate Heat Transfer, Fluid–Solid Interaction[2]		—	3D, 2D, 2D axisymmetric	stationary; time dependent;
Fluid–Solid Interaction, Viscoelastic Flow[2,10]		—	3D, 2D, 2D axisymmetric	time dependent; time dependent with phase initialization
Fluid–Solid Interaction, Two-Phase Flow, Phase Field[2,6]		—	3D, 2D, 2D axisymmetric	time dependent; time dependent with phase initialization

PHYSICS INTERFACE		ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
	Solid–Thin-Film Damping[2]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, modal reduced-order model; time dependent; response spectrum; random vibration (PSD)
	Shell–Thin-Film Damping[2]		—	3D, 2D axisymmetric	stationary; eigenfrequency; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, modal reduced-order model; time dependent; response spectrum; random vibration (PSD)

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Structural Mechanics				
Solid Mechanics[1]		solid	3D, 2D, 2D axisymmetric, 1D, 1D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; mode analysis; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; frequency domain, AWE reduced-order model; response spectrum; random vibration (PSD); linear buckling; bolt pretension
Solid Mechanics, Explicit Dynamics		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; explicit dynamics; bolt pretension

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Shell		shell	3D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; response spectrum; random vibration (PSD); linear buckling
Plate		plate	2D	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; response spectrum; random vibration (PSD); linear buckling

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Beam		beam	3D, 2D	stationary; eigenfrequency; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, modal reduced-order model; time dependent; response spectrum; random vibration (PSD); linear buckling
Beam Cross Section		bcs	3D, 2D	stationary
Truss		truss	3D, 2D	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; response spectrum; random vibration (PSD); linear buckling
Truss, Explicit Dynamics		truss	3D, 2D	stationary; eigenfrequency; time dependent; frequency domain; explicit dynamics

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Wire		wire	3D, 2D	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; response spectrum; random vibration (PSD); linear buckling
Membrane		mbrn	3D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; frequency domain, modal reduced-order model; response spectrum; random vibration (PSD)

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Pipe Mechanics		pipem	3D, 2D	stationary; eigenfrequency; time dependent, modal; time dependent, modal reduced-order model; frequency domain; frequency domain, modal; frequency domain, modal reduced-order model; time dependent; response spectrum; random vibration (PSD)
Phase Field Damage[2,11]		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Phase Field Damage, Explicit Dynamics[2,11]		—	3D, 2D, 2D axisymmetric	stationary; time dependent; explicit dynamics
Thermal–Structure Interaction				
Thermal Stress, Solid[2]		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Thermal Stress, Shell[2,4]		—	3D, 2D axisymmetric	stationary; time dependent
Thermal Stress, Membrane[2,4]		—	3D, 2D axisymmetric	stationary; time dependent
Joule Heating and Thermal Expansion[2]		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Piezoelectricity				

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Piezoelectricity, Solid[2]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; small-signal analysis, frequency domain; linear buckling
Piezoelectricity, Layered Shell[2,5]		—	3D	stationary; eigenfrequency; time dependent; frequency domain
Magnetomechanics				
Piezomagnetism[2,3]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Nonlinear Magnetostriction[2,3]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetomechanics, Solids[2,3]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Magnetomechanics, No Currents, Solids[2,3]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetomechanics, Shell[2,3]		—	3D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetomechanics, No Currents, Shell[2,3]		—	3D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetomechanics, Membrane[2,3]		—	3D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetomechanics, No Currents, Membrane[2,3]		—	3D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Magnetic–Elastic Interaction in Rotating Machinery[2,3]		—	3D, 2D	stationary; time dependent; frequency domain;
Electrostriction				

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Ferroelectroelasticity[2,9]		—	3D, 2D, 2D axisymmetric	stationary; time dependent; frequency domain
Electrostriction[2,9]		—	3D, 2D, 2D axisymmetric	stationary; time dependent; frequency domain
Piezoresistivity				
Piezoresistivity, Shell[2,12]		—	3D	stationary; time dependent; time dependent, modal; frequency domain; frequency domain, modal; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Piezoresistivity, Layered Shell[2,5,12]		—	3D	stationary; time dependent; time dependent, modal; frequency domain; frequency domain, modal; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Poroelasticity				
Unsaturated Poroelasticity[2,8]		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Mathematics				
Moving Interface				
Phase Field in Solids		pfs	3D, 2D, 2D axisymmetric	stationary, time dependent

[1] This physics interface is included with the core COMSOL Multiphysics software but has added functionality for this module.

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
[2] This physics interface is a predefined multiphysics coupling that automatically adds all the physics interfaces and coupling features required.				
[3] Requires the addition of the AC/DC Module.				
[4] Requires the addition of the Heat Transfer Module.				
[5] Requires the addition of the Composite Materials Module.				
[6] Requires the addition of the CFD Module, or the Polymer Flow, or the Microfluidics Module.				
[7] Requires the addition of the Pipe Flow Module.				
[8] Requires the addition of the Porous Media Flow Module.				
[9] Requires the addition of the AC/DC Module or the MEMS Module.				
[10] Requires the addition of the Polymer Flow Module.				
[11] Requires the addition of the Nonlinear Structural Materials Module or the Geomechanics Module.				
[12] Requires the addition of the MEMS Module.				

Common Physics Interface and Feature Settings and Nodes

There are several common settings and sections available for the physics interfaces and feature nodes. Some of these sections also have similar settings or are implemented in the same way no matter the physics interface or feature being used. There are also some physics feature nodes that display in COMSOL Multiphysics.

In each module's documentation, only unique or extra information is included; standard information and procedures are centralized in the *COMSOL Multiphysics Reference Manual*.

	In the *COMSOL Multiphysics Reference Manual*, see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Where Do I Access the Documentation and Application Libraries?

A number of online resources have more information about COMSOL, including licensing and technical information. The electronic documentation, topic-based (or context-based) help, and the Application Libraries are all accessed through the COMSOL Desktop.

	If you are reading the documentation as a PDF file on your computer, the blue links do not work to open an application or content referenced in a different guide. However, if you are using the Help system in COMSOL Multiphysics, these links work to open other modules, application examples, and documentation sets.

THE DOCUMENTATION AND ONLINE HELP

The *COMSOL Multiphysics Reference Manual* describes the core physics interfaces and functionality included with the COMSOL Multiphysics license. This book also has instructions on how to use COMSOL Multiphysics and how to access the electronic Documentation and Help content.

Opening Topic-Based Help

The Help window is useful as it is connected to the features in the COMSOL Desktop. To learn more about a node in the Model Builder, or a window on the Desktop, click to highlight a node or window, then press F1 to open the Help window, which then

displays information about that feature (or click a node in the Model Builder followed by the **Help** button (?). This is called *topic-based* (or *context) help*.

Win

To open the **Help** window:

- In the **Model Builder**, **Application Builder**, or **Physics Builder**, click a node or window and then press F1.
- On any toolbar (for example, **Home**, **Definitions**, or **Geometry**), hover the mouse over a button (for example, **Add Physics** or **Build All**) and then press F1.
- From the **File** menu, click **Help** (?).
- In the upper-right corner of the COMSOL Desktop, click the **Help** (?) button.

Mac

Linux

To open the **Help** window:

- In the **Model Builder** or **Physics Builder**, click a node or window and then press F1.
- In the main toolbar, click the **Help** (?) button.
- From the main menu, select **Help** > **Help**.

Opening the Documentation Window

Mac

Linux

To open the **Documentation** window:

- Press Ctrl+F1.
- In the main toolbar, click the **Documentation** ( ) button.
- From the main menu, select **Help** > **Documentation**.

THE APPLICATION LIBRARIES WINDOW

Each model or application includes documentation with the theoretical background and step-by-step instructions to create a model or application. The models and applications are available in COMSOL Multiphysics as MPH files that you can open for further investigation. You can use the step-by-step instructions and the actual models as templates for your own modeling. In most models, SI units are used to describe the relevant properties, parameters, and dimensions, but other unit systems are available.

Once the Application Libraries window is opened, you can search by name or browse under a module folder name. Click to view a summary of the model or application and its properties, including options to open it or its associated PDF document.

Opening the Application Libraries Window

To open the **Application Libraries** window ( ):

CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at info@comsol.com.

COMSOL ACCESS AND TECHNICAL SUPPORT

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and a case number will be sent to you by email. You can also access technical support, software updates, license information, and other resources by registering for a COMSOL Access account.

COMSOL ONLINE RESOURCES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
COMSOL Access	www.comsol.com/access
Support Center	www.comsol.com/support
Product Download	www.comsol.com/product-download
Product Updates	www.comsol.com/product-update
COMSOL Blog	www.comsol.com/blogs
Discussion Forum	www.comsol.com/forum
Events	www.comsol.com/events
COMSOL Application Gallery	www.comsol.com/models
COMSOL Video Gallery	www.comsol.com/videos
Learning Center	www.comsol.com/support/learning-center
Support Knowledge Base	www.comsol.com/support/knowledgebase

Overview of the User's Guide

The *Structural Mechanics Module User's Guide* gets you started with modeling using COMSOL Multiphysics. The information in this guide is specific to this module. Instructions how to use COMSOL in general are included with the *COMSOL Multiphysics Reference Manual*.

As detailed in the section Where Do I Access the Documentation and Application Libraries?, this information can also be searched from the **Help** menu in COMSOL Multiphysics.

TABLE OF CONTENTS, GLOSSARY, AND INDEX

To help you navigate through this guide, see the Contents, Glossary, and Index.

MODELING WITH THE STRUCTURAL MECHANICS MODULE

The Structural Mechanics Modeling chapter gives you an insight on how to approach the modeling of various structural mechanics problems.

STRUCTURAL MECHANICS THEORY

The Structural Mechanics Theory chapter introduces the general theory on which the physics interfaces in the Structural Mechanics Module are based.

THE SOLID MECHANICS INTERFACE

The Solid Mechanics chapter describes The Solid Mechanics Interface, which is used to model 3D solids, plane strain and plane stress 2D and 1D models, and 2D axisymmetric models.

THE SHELL AND PLATE INTERFACES

The Shell and Plate chapter describes The Shell and Plate Interfaces, which are used to model thin 3D structures (shell) and out-of-plane loaded plates (plate). The underlying theory is described in Theory for the Shell and Plate Interfaces.

THE BEAM INTERFACE

The Beam chapter describes The Beam Interface, which contains Euler (Euler–Bernoulli) and Timoshenko beams for modeling slender 3D and 2D structures. Typical examples are frameworks and latticeworks. The underlying theory for the physics interface is described in Theory for the Beam Interface.

THE BEAM CROSS SECTION INTERFACE

The Beam Cross Section chapter describes The Beam Cross Section Interface, which is used for computing cross section properties for beams. It can also be used for a detailed evaluation of stresses in a beam when the section forces to which it is subjected are known. The first section discusses Using the Beam Cross Section Interface, and the underlying theory is described in Theory for the Beam Cross Section Interface.

THE TRUSS INTERFACE

The Truss chapter describes The Truss Interface, which models slender 3D and 2D structures with components capable to withstand axial forces only. Typical applications are latticeworks, but it can also be used for modeling cables. In the section Modeling with Truss Elements, you will find a discussion about how to set up models using this interface. The underlying theory for the physics interface is described in Theory for the Truss Interface.

THE MEMBRANE INTERFACE

The Membrane chapter describes The Membrane Interface, which can be used for modeling prestressed membranes, cladding on solids, and balloons, for example. The underlying theory for the physics interface is also included in Theory for the Membrane Interface.

THE WIRE INTERFACE

The Wire chapter describes The Wire Interface, which is mainly intended for modeling pretensioned or sagging wires or cables. The underlying theory for the physics interface is also included in Theory for the Wire Interface.

THE PIPE MECHANICS INTERFACE

The Pipe Mechanics chapter describes The Pipe Mechanics Interface, which is used for studying the stresses and deformations in piping systems. The underlying theory for the physics interface is also included in Theory for the Pipe Mechanics Interface.

THE PHASE FIELD IN SOLIDS INTERFACE

The Phase Field in Solids chapter describes The Phase Field in Solids Interface, which is used to determine the degradation of material properties in the damaged regions. The underlying theory for the physics interface is also included in Theory for the Phase Field in Solids Interface.

THE TRANSPORT IN SOLIDS INTERFACE

The Transport in Solids chapter describes The Transport in Solids Interface, which can be used to model transport phenomena in solids. The underlying theory for the physics interface is also included in Theory for the Transport in Solids Interface.

THE MULTIPHYSICS INTERFACES

The Multiphysics Interfaces and Couplings chapter describes these physics interfaces found under the Structural Mechanics branch when adding a physics interface:

- The Thermal Stress, Solid Interface combines a Solid Mechanics interface with a Heat Transfer interface. The coupling appears on the domain level, where the temperature from the Heat Transfer interface acts as a thermal load for the Solid Mechanics interface, causing thermal expansion.
- The Thermal Stress, Shell Interface combines a Shell interface with a Heat Transfer in Shells interface. The coupling appears on the boundary level, where the temperature from the Heat Transfer in Shells interface acts as a thermal load for the Shell interface, causing thermal expansion.
- The Thermal Stress, Layered Shell Interface combines a Layered Shell interface with a Heat Transfer in Shells interface. The coupling appears on the boundary level, where the temperature from the Heat Transfer in Shells interface acts as a thermal load for the Layered Shell interface, causing thermal expansion.
- The Thermal Stress, Membrane Interface combines a Membrane interface with a Heat Transfer in Shells interface. The coupling appears on the boundary level, where the temperature from the Heat Transfer in Shells interface acts as a thermal load for the Membrane interface, causing thermal expansion.
- The Joule Heating and Thermal Expansion Interface combines solid mechanics using a thermal linear elastic material with an electromagnetic Joule heating model. This is a multiphysics combination of solid mechanics, electric currents, and heat transfer for modeling of, for example, thermoelectromechanical (TEM) applications.
- The Piezoelectricity, Solid Interface combines a Solid Mechanics interface with an Electrostatics interface. Piezoelectric materials in 3D, 2D plane strain and plane stress, and 2D axial symmetry can be modeled.
- The Piezoelectricity, Layered Shell Interface combines a Layered Shell interface with an Electric Currents in Layered Shells interface. This makes it possible to model piezoelectric effects in thin layered structures.

- The Piezoelectricity, Layered Shell Interface combines a Layered Shell interface with an Electric Currents in Layered Shells interface. This makes it possible to model piezoelectric effects in thin layered structures.
- The Electrostriction Interface combines a Solid Mechanics with an Electrostatics interface. Using this interface, you can solve problems where strains are caused by electrostrictive effects.
- The Ferroelectroelasticity Interface combines a Solid Mechanics with an Electrostatics interface. Using this interface, you can solve problems involving ferroelectric materials, for example within nonlinear piezoelectricity.
- Hygroscopic Swelling combines a Solid Mechanics with a Magnetic Fields interface. Using this interface, you can solve problems in the magnetostrictive field with linear as well as nonlinear material models.
- The Fluid–Solid Interaction Interface combines fluid flow with the Solid Mechanics interface to capture the interaction between the fluid and the solid in a situation where the fluid domain has significant deformation. The solid material exists on domains which are adjacent to the fluid.
- The Fluid–Shell Interaction Interface combines fluid flow with the Shell interface to capture the interaction between the fluid and the solid in a situation where the fluid domain has significant deformation. The shell is modeled on the boundary of the fluid.
- The Fluid–Membrane Interaction Interface combines fluid flow with the Membrane interface to capture the interaction between the fluid and the membrane in a situation where the fluid domain has significant deformation. The membrane is modeled on the boundary of the fluid.
- The Fluid–Solid Interaction, Fixed Geometry Interface combines fluid flow with the Solid Mechanics interface to capture the interaction between the fluid and the solid in a situation where the fluid domain can be considered to be nondeforming. The solid material exists on domains which are adjacent to the fluid.
- The Fluid–Shell Interaction, Fixed Geometry Interface combines fluid flow with the Shell interface to capture the interaction between the fluid and the solid in a situation where the fluid domain can be considered to be nondeforming. The shell is modeled on the boundary of the fluid
- The Fluid–Membrane Interaction, Fixed Geometry Interface combines fluid flow with the Membrane interface to capture the interaction between the fluid and the membrane in a situation where the fluid domain can be considered to be nondeforming. The membrane is modeled on the boundary of the fluid.

- The Fluid–Pipe Interaction, Fixed Geometry Interface combines flow computed using the Pipe Flow interface with structural analysis in the Pipe Mechanics interface. Different types of fluid loads are transferred to the structural analysis.
- The Fluid–Solid Interaction, Conjugate Heat Transfer Interface combines fluid flow with the Solid Mechanics interface and the Heat Transfer in Solids and Fluids interface. It combines fluid-structure interaction modeling with a nonisothermal flow. Heat transfer is considered both in the fluid and in the solid in order to capture thermal expansion effects.
- The Fluid–Solid Interaction, Two-Phase Flow, Phase Field Interface combines two-phase fluid flow with the Solid Mechanics interface to capture the interaction between the fluid and the solid in a situation where the fluid domain has significant deformation. The solid material exists on domains which are adjacent to the fluid.
- The Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry Interface combines two-phase fluid flow with the Solid Mechanics interface to capture the interaction between the fluid and the solid in a situation where the fluid domain can be considered to be nondeforming. The solid material exists on domains which are adjacent to the fluid.
- The Fluid–Solid Interaction, Viscoelastic Flow Interface combines the Viscoelastic Flow interface with a Deforming Domain feature and the Solid Mechanics interface to capture the interaction between a viscoelastic fluid and solids in a situation where the fluid domain has significant deformation.
- The Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry Interface combines the Viscoelastic Flow interface with a Deforming Domain feature and the Solid Mechanics interface to capture the interaction between a viscoelastic fluid and solids in a situation where the fluid domain can be considered to be nondeforming.
- The Solid–Thin-Film Damping Interface combines the Thin-Film Flow interface and the Solid Mechanics interface to model phenomena where a thin-film fluid and a deformable solid affect each other. The fluid can be either a liquid or a gas, with the possibility to include cavitation in liquids.
- The Shell–Thin-Film Damping Interface combines the Thin-Film Flow interface and the Shell interface to model phenomena where a thin-film fluid and a deformable shells affect each other. The fluid can be either a liquid or a gas, with the possibility to include cavitation in liquids.
- The Poroelasticity, Solid Interface combines a transient formulation of Darcy's law with a quasistatic formulation of Solid Mechanics. The coupling occurs on the domain level, where the pore pressure from the Darcy's Law interface acts as a load

for the Solid Mechanics interface, causing swelling or shrinking. Changes in volumetric strain affect the pore space, acting as a mass source or sink in Darcy's Law.

- The Poroelasticity, Layered Shell Interface combines the Layered Darcy's Law formulation with the Layered Shell interface for structural mechanics. This multiphysics interface is only available for 3D simulations and it can be active on boundaries where a layered material is present.
- The Unsaturated Poroelasticity Interface combines a transient formulation of Moisture Transport in Solids with a quasistatic formulation of Solid Mechanics. The coupling occurs on the domain level, where the pore pressure from the Moisture Transport in Solids interface acts as a load for the Solid Mechanics interface, causing swelling or shrinking. Changes in volumetric strain affect the pore space, impacting the transport of moist air and liquid water in the porous solid.
- The Phase-Field Damage Interface combines the Solid Mechanics and Phase Field in Solids interfaces to model the evolution of damage and cracks in deforming solids. The coupling occurs on the domain level, where the driving force for damage depends on the state of deformation in the solid, which governs the evolution of the phase field. In turn, the phase field determines the degradation of the solid properties in the damaged regions.

2

Structural Mechanics Modeling

The goal of this chapter is to give you an insight on how to approach the modeling of various structural mechanics problems.

Some physics interfaces and features discussed in this chapter are only available with certain products. For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/.

In this chapter:

- Study Types
- Selecting the Physics Interface
- Selecting Discretization
- Using Reduced Integration
- Modeling with Plates and Shells
- Modeling with Layered Shells
- Modeling with Membrane Elements
- Modeling with Beam Elements
- Modeling with Truss Elements
- Modeling Wires and Cables
- Modeling with the Phase Field in Solids Interface
- Modeling with the Transport in Solids Interface
- Coupling Different Element Types
- Modeling Embedded Structures and Reinforcements
- Applying Loads
- Defining Constraints
- Calculating Reaction Forces
- Introduction to Material Models
- Modeling Piezoelectric Problems
- Modeling Electrostrictive and Ferroelectroelastic Materials
- Modeling Magnetostrictive Materials
- Mechanical Damping and Losses
- Modeling Geometric Nonlinearity
- Contact Modeling
- Activating and Deactivating Material
- Springs and Dampers
- Defining Multiphysics Models
- Thermally Coupled Problems
- Fluid–Structure Interaction
- Component Mode Synthesis
- Computing Mass Properties
- Effective Properties of Periodic Structures
- Modeling Bolts, Fasteners, and Welds
- Modeling Thin Layers
- Modeling Cracks
- Buckling Analysis
- Performing an Explicit Dynamics Analysis
- Performing a Random Vibration Analysis
- Performing a Response Spectrum Analysis
- Performing an Inertia Relief Analysis
- Stress Linearization
- Solver Settings for Structural Mechanics
- Result Presentation
- Part Libraries
- Estimation of Material Parameters

Study Types

Introduction

In this section, you will find information about when and how to apply the study types which are available for structural mechanics problems:

- Stationary Analysis
- Eigenfrequency Analysis
- Mode Analysis
- Time-Domain Analysis
- Frequency-Domain Analysis
- Mode Superposition
- Harmonic Perturbation
- Linearized Buckling Analysis
- Bolt Pretension Study
- Random Vibration (PSD) Study
- Response Spectrum Analysis Study
- Inertia Relief Study

For general information about study types and solvers, see Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

Stationary Analysis

You can consider a structural mechanics problem as stationary if the following two criteria are fulfilled:

- The loads vary so slowly that inertial forces are negligible. Problems of this type are referred to as *quasi static*.
- There are no explicit time dependencies in the material model. Viscoelasticity and creep, for example, have such time dependencies.

To perform this type of analysis, you use a **Stationary** study step.

In many cases, there is a variation in the load, even though the solution for each value of the load can be considered as stationary. There are three conceptually different cases:

- The load values are independent; it is just a number of different load cases you want to compute. The load case handling functionality described in Load Cases is well suited for this purpose.
- You want to study a nonlinear problem where the solution is path dependent, or where the load must be increased in small increments in order to obtain a converged solution. In this case you should use the parametric continuation solver. Create a parameter under **Global Definitions** > **Parameters**, which you use to control the variation of the load. Then select **Auxiliary sweep** under **Study Extensions** in the settings for the **Stationary** solver. In the table for the auxiliary sweep parameters, add the load controlling parameter, and define its range of variation.
- In a multiphysics problem, another physical quantity might be truly time dependent but on a time scale that is "slow" from the structural mechanics point of view. This is usually the case with, for example, problems coupled to heat transfer or diffusion. If the problem also is such that the structural deformations do not affect the other physics, it will be unnecessarily expensive to solve also the structural problem in the time domain, irrespective of whether it is linear or nonlinear. In this situation, you should first solve the other physics in a time-dependent study and then the structural mechanics problem in a subsequent stationary study step using the time `t` as the parameter in the auxiliary sweep.

For an example of how to set up such a study sequence, see *Fluid–Structure Interaction in a Network of Blood Vessels*: Application Library path **Structural_Mechanics_Module/Fluid-Structure_Interaction/blood_vessel**.

CONSTRAINTS

A stationary problem is solvable only if the structure is sufficiently constrained. There must not be any possible rigid body modes.

For a more detailed discussion about sufficient constraints, see Rigid Body Motion.

Eigenfrequency Analysis

An **Eigenfrequency** study solves for the eigenfrequencies (natural frequencies) and the shapes of the corresponding eigenmodes.

When performing an eigenfrequency analysis, you can specify whether to look at the mathematically more fundamental eigenvalue, λ, or the eigenfrequency, f, which is more commonly used in a structural mechanics context. The relation between the two is

$$f = -\frac{\lambda}{2\pi i}$$

where i is the imaginary unit.

The undamped eigenvalue problem is commonly written as

$$[\mathbf{K} - \omega^2\mathbf{M}]\mathbf{u} = 0$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{M}$ is the mass matrix, $\mathbf{u}$ is the eigenmode displacement vector, and $\omega = 2\pi f$ is the angular frequency. If damping is present, the eigenvalue equation is expanded to

$$[\mathbf{K} + i\omega\mathbf{C} - \omega^2\mathbf{M}]\mathbf{u} = 0 \tag{2-1}$$

where $\mathbf{C}$ is the viscous damping matrix, and $\mathbf{K}$ can be complex-valued.

Because only the shape, and not the size, of the modes (eigenvectors) have physical significance, the computed modes can be scaled arbitrarily. You can select the method for scaling in the **Eigenvalue Solver** node of the solver sequence. If **Scaling of eigenvectors** is set to **Mass matrix**, the eigenmodes $\mathbf{u}$ are orthogonalized with respect to the mass matrix M so that

$$\mathbf{u}_i^T M \mathbf{u}_i = 1 \tag{2-2}$$

This is a common choice for the scaling of eigenvectors within the structural mechanics field. The choice of eigenvector scaling does not affect for example the results of a

subsequent mode superposition analysis, but it will affect the interpretation of an exported modal representation of the system.

- There are some situations where nonlinear expressions are used for evaluation of result quantities. Such expressions will only be correct if the scale of the eigenmodes is small enough so that any nonlinearities are negligible. For this reason, the default scaling uses a small number. The maximum value of an eigenmode is set to 10^{-6} times the size of the bounding box of the geometry.
- The maximum value scaling checks each degree of freedom separately. This means that in a pure solid mechanics problem, the norm of the displacement of a mode shape can exceed the given scale somewhat.
- In a multiphysics problem, it is possible that the largest values in the eigenmode vector represent another field, for example, electric potential. In that case, the largest displacement components will be smaller than the specified number.

MODAL PARTICIPATION FACTORS

Modal (or mass) participation factors are useful tools when working with the modal representation of a structure. Through them, you can get the following information:

- The fraction of the total mass of a structure that a certain number of modes represent is one result. This can be important when judging if a set of modes forms a good enough base for a mode superposition.
- The main direction of vibration for a certain mode can be seen from the relation between the participation factors.
- When you have a large set of modes, an examination of the participation factors can give information about the dominant modes. This may then be used for a reduction of the size of subsequent modal-based studies.

To compute modal participation factors, a Participation Factors node must be present under **Definitions** in the current component. When you add an **Eigenfrequency** study from the **Add Study** window, such a node is automatically created.

You can also add a **Participation Factors** node manually under **Definitions** > **Physics Utilities**. If you do that after an eigenfrequency study has been run, you need to do an **Update Solution** in order to get access to the variables containing the participation factors.

The modal participation factors are available as global variables, and these can, for example, be displayed in a table using a **Global Evaluation** node under **Derived Values** in the **Results** branch. The participation factor results are available as predefined variables in the **Definitions** submenu for the component. In Table 2-1, the variables created from a **Participation Factors** node is listed (assuming the default tag `mpf1`).

TABLE 2-1: PARTICIPATION FACTOR VARIABLES.

VARIABLE	DESCRIPTION
`mpf1.mass`	Total mass
`mpf1.CMJ`	Center of mass, J coordinate
`mpf1.mEffLJ`	Effective mass, translation along J-axis
`mpf1.mEffRJ`	Effective mass, rotation around J-axis
`mpf.pfLJ`	Participation factor, translation along J-axis
`mpf.pfRJ`	Participation factor, rotation around J-axis
`mpf.pfLnormJ`	Normalized participation factor, translation along J-axis
`mpf.pfRnormJ`	Normalized participation factor, rotation around J-axis

The normalized participation factors are those that would be obtained if mass matrix scaled eigenmodes would have been used.

If a **Participation Factors** node is present in the model when an eigenfrequency study is run, an evaluation group named **Participation Factors** is automatically generated. It contains a table with the translational and rotational modal participation factors for all computed eigenfrequencies.

- If you would compute all eigenmodes of a structure, and sum all modal masses, they will usually not exactly match the total mass of the structure. The reason is that any mass which is associated with constrained degrees of freedom is lost. This effect is discretization dependent. The mass lost is a fraction of the mass of the elements having constrained nodes.
- The computation of participation factors is a feature at the component level. For a case where there are couplings between two or more components that affect the eigenvalue problem, an evaluation of the participation factors and modal masses may give unpredictable results.

For an example showing how to compute modal participation factors and modal mass, see *In-Plane Framework with Discrete Mass and Mass Moment of Inertia*: Application Library path **Structural_Mechanics_Module/Verification_Examples/inplane_framework_freq**.

In the *COMSOL Multiphysics Reference Manual*:

- Eigenvalue Solver
- Studies and Solvers
- Postprocessing of Eigenmodes
- Derived Values, Evaluation Groups, and Tables

In the theory chapter of the *Structural Mechanics User's Guide*:

- Modal Participation Factors

It is possible to compute eigenfrequencies and eigenmodes for structures which are not fully constrained. Such eigenmodes are sometimes referred to as *free-free* modes. For each possible rigid body mode, there is one eigenvalue which in theory is zero. The number of possible rigid body modes for different geometrical dimensions is shown in the table below.

TABLE 2-2: NUMBER OF POSSIBLE RIGID BODY MODES.

DIMENSION	NUMBER OF RIGID BODY MODES
3D	6 (3 translations + 3 rotations)
2D axisymmetric	1 (Z-direction translation)
2D (solid, beam, truss)	3 (2 translations + 1 rotation)
2D (plate)	3 (1 translation + 2 rotations)

The computed rigid body modes will in general not be recognizable as having pure translation or rotation. Rather, they will contain linear combinations of all the fundamental rigid body motions.

In a piezoelectric model, one more zero eigenfrequency could appear if you have not set a reference value for the electric potential.

In practice, the natural frequencies of the rigid body modes are not computed as exactly zero, but can appear as small numbers which may even be negative or complex. If rigid body modes are present in the model, then it is important to use a nonzero

value in the **Search for eigenfrequencies around** text field in the settings for the Eigenfrequency study step. The value should reflect the order of magnitude of the first important nonzero eigenfrequency.

For an example showing an eigenfrequency computation in a model having a rigid body mode, see *Eigenfrequency Analysis of a Free Cylinder*: Application Library path **Structural_Mechanics_Module/Verification_Examples/free_cylinder**.

DAMPING

If any type of damping is included in the model, an eigenfrequency solution automatically returns the damped eigenvalues. The eigenfrequencies and, in general, also the mode shapes are complex in this case. A complex-valued eigenfrequency can be interpreted so that the real part represents the actual frequency, and the imaginary part represents the damping. The damping ratio of the corresponding eigenmode can be defined as

$$\zeta_i = \frac{\text{imag}(\omega_i)}{|\omega_i|} \approx \frac{\text{imag}(\omega_i)}{\text{real}(\omega_i)}$$

where the approximate expression is valid with high accuracy (within 2%) as long as the damping is less than 0.2.

If the damping ratio is high, more than 0.6 or so, then the default ARPACK eigenvalue solver will show better convergence if you set **Search method around shift** to **Larger real part**. In particular, Rayleigh damping can cause high damping ratios for higher eigenmodes, since the damping asymptotically is proportional to the frequency.

In a complex mode shape, there are phase shifts between different parts of the structure, so that not all points reach the maximum at the same time under *free vibration*.

Some damping types will still give real-valued eigenmodes; this is the case for Rayleigh damping and loss factor damping.

If an eigenfrequency study is performed as a first step of a mode superposition analysis, then all features that supply damping should be disabled in this step. This can be done in the **Physics and Variables Selection** section in the settings for the **Eigenfrequency** study step.

The damping is taken into account in the mode superposition studies.

See also the discussion under Mode Superposition.

PRESTRESSED ANALYSIS

In a loaded structure, the natural frequencies may be shifted due to *stress stiffening*.

To do a prestressed analysis, **Include geometric nonlinearity** must be selected in the **Eigenfrequency** study step. This is automatic when you add the Eigenfrequency, Prestressed study type.

The prestress loading can include a contact analysis, in which case the subsequent eigenfrequency analysis provide a linearization around the current contact state.

Prestressed Structures

FREQUENCY DEPENDENT MATERIAL PROPERTIES

If the material data (stiffness or damping) is frequency dependent, the eigenvalue problem will become nonlinear. This can, for example, occur for some viscoelastic materials. In this case, the eigenvalue equation Equation 2-1 becomes

$$[\mathbf{K}(\omega) + i\omega\mathbf{C}(\omega) - \omega^2\mathbf{M}]\mathbf{u} = 0 \tag{2-3}$$

The eigenvalue solver as such assumes that the matrices involved are constant, so they must be evaluated at a certain frequency, the *linearization point*.

$$[\mathbf{K}(\omega_L) + i\omega\mathbf{C}(\omega_L) - \omega^2\mathbf{M}]\mathbf{u} = 0 \tag{2-4}$$

In order to get a correct solution to Equation 2-3, the linearization point ω_L must be close to the actual eigenvalue ω. This is in general possible only for one single

eigenfrequency at a time. You must solve this problem either by manual iteration, or by using some type of scripting, for example through a model method.

Eigenmodes of a Viscoelastic Structural Damper: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_eigenmodes**.

The built-in viscoelastic materials use a special formulation that do not generate a nonlinear eigenvalue problem, as long fractional derivatives are not used.

Mode Analysis

The **Mode Analysis** study type ( ) is available with the Solid Mechanics interface in 2D plane strain.

Elastic waves can propagate over large distances in structures like rails and pipes, sometimes referred to as *waveguides*. After some distance of propagation in a waveguide of uniform cross section, such guided waves can be described as a sum of just a few discrete *propagating modes*, each with its own shape and phase speed. The equation governing these modes can be obtained as a spatial Fourier transform of the linearized time-harmonic equation in the waveguide axial z direction or by inserting the assumption that the mode is harmonic in space,

$$u = ue^{-ik_z z}$$

and eliminating all out-of-plane z dependence.

Similar to the full time-harmonic equation, the transformed equation can be solved at a given frequency with a nonzero excitation for most axial wave numbers k_z. But at certain discrete values the equation breaks down. These values are the propagation constants or wave numbers of the propagating or evanescent waveguide modes. The eigenvalue solver can solve for these propagation constants together with the corresponding mode shapes.

The propagating wave number is a function of the frequency. The relation between the two is commonly referred to as a *dispersion curve*.

The most common use for mode analysis is to define sources for a subsequent time-harmonic simulation. If there is a component with one or more waveguide connections, its behavior can be described by simulating its response to the discrete set of propagating modes on the waveguide opening cross sections.

- Out-of-Plane Waves in the *Structural Mechanics Theory* chapter.
- Studies and Solvers and Mode Analysis in the *COMSOL Multiphysics Reference Manual.*

Time-Domain Analysis

There are two classes of problems where a stationary solution cannot be used:

- When the inertial forces no longer are negligible, the full problem as given by Newton's second law must be solved.
- When there are time dependencies in a material model, as for creep or viscoelasticity.

The most general way of handling time-dependent problems is to use a **Time Dependent** study. In this type of analysis, you can incorporate any type of nonlinearity, and there are no limitations on the time-dependence of the loads.

A time-domain solution can be preceded by a stationary study, if, for example, prestress effects are needed. This is also an efficient way of establishing correct initial conditions if there are static loads, or if some load should start at a non-zero value.

For a linear problem including inertia, using the mode superposition method is often much more efficient than using the standard direct method.

SOLVER SELECTION

The two classes of dynamic problems presented above have quite different properties. The inertial forces in the full structural dynamics problem contain second-order time derivatives of the displacements, whereas creep and viscoelasticity only have first-order derivatives. The physical and numerical properties of these equations differ significantly.

There are two general implicit solvers for time-dependent problems in COMSOL Multiphysics.

- The *generalized alpha* method, which is recommended for structural dynamics problems. This is the default solver if **Structural Transient Behavior** is set to **Include inertial terms** in the physics interface settings.
- The BDF method, which is recommended for first-order problems. This is the default solver if **Structural Transient Behavior** is set to **Quasistatic** in the physics interface settings

A structural dynamics problem can also be solved using explicit time-stepping, so-called *explicit dynamics*. COMSOL Multiphysics supports several explicit methods where the *Verlet* method is of main interest for structural mechanics. This is the default solver when using the **Explicit Dynamics** study step to set up a structural problem.

In the *COMSOL Multiphysics Reference Manual*:

- Time-Dependent Solver
- Studies and Solvers
- Explicit Dynamics

See Performing an Explicit Dynamics Analysis for more details solve a time-domain analysis using explicit time-stepping.

- *Bracket — Transient Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_transient**.
- *Bracket — Explicit Dynamics*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_explicit**
- *Stress Relaxation of a Viscoelastic Tube*: Application Library path **Structural_Mechanics_Module/Material_Models/viscoelastic_tube**.

Frequency-Domain Analysis

In a frequency-domain analysis, you study the response to a harmonic steady state excitation for certain frequencies. Such a steady state can prevail once all transient effects have been damped out.

The response must be linear, so that the single frequency harmonic excitation gives a pure harmonic response with the same frequency. The model may, however, contain

nonlinearities. The harmonic response is computed around a certain linearization point. In such a case, the frequency-domain analysis can be considered as a very small perturbation around that linearization point.

Harmonic Perturbation

All loads and responses in a frequency-domain analysis are in general complex-valued quantities. If all loads do not have the same phase, you can describe the phase of a certain load in two ways:

- Add a **Phase** subnode to the load, in which you directly give the phase angle.
- Enter the load as a complex value, for example as `100[N]*(1+0.3*i)/sqrt(1+0.3^2)`.

Most results from a frequency-domain analysis are complex valued. In many result evaluation nodes, the real value of any result quantity will be shown. Assuming that you want to display for example the displacement in the x direction, `u`, you have the following options:

- Plot `u` or `real(u)`. This gives the displacement at current (default zero) phase angle.
- Plot `imag(u)`. This gives the displacement at a phase angle shifted 90° from the current value.
- Plot `abs(u)`. This gives the amplitude of the displacement.
- Plot `arg(u)`. This gives the phase angle of the displacement.

The reference phase, with respect to which the results above are reported can be entered in the settings for the dataset as well as in the settings for a certain result node, like a surface plot.

Result quantities that are nonlinear in terms of the displacements, such as principal stresses, should be interpreted with great care in frequency domain. They will in general not be harmonic, so the information about amplitude and phase is not reliable.

Some extra variables for postprocessing are created in a frequency-domain analysis. As an example, in a Solid Mechanics interface with the name `solid`, the following variables are defined:

- `solid.disp` — norm of displacement (at current phase angle)
- `solid.vel` — norm of velocity (at current phase angle)
- `solid.acc` — norm of acceleration (at current phase angle)
- `solid.disp_rms` — RMS displacement over a cycle
- `solid.vel_rms` — RMS velocity over a cycle
- `solid.acc_rms` — RMS acceleration over a cycle
- `solid.uAmpX` — amplitude of displacement in the *X* direction
- `solid.uAmp_tX` — amplitude of velocity in the *X* direction
- `solid.uAmp_ttX` — amplitude of acceleration in the *X* direction
- `solid.uPhaseX` — phase of X displacement, in radians
- `solid.uPhase_tX` — phase of X velocity, in radians
- `solid.uPhase_ttX` — phase of X displacement, in radians
- `solid.mises` — von Mises equivalent stress at current phase angle.
- `solid.mises_peak` — maximum von Mises equivalent stress over a cycle. In a prestressed analysis, this is the maximum of the perturbation stress.

The components in other coordinate directions are obtained by replacing `X` by another coordinate name.

PRESTRESSED ANALYSIS

The shift in the natural frequencies in a prestressed structure may have a significant effect on the frequency response. This is particularly important when the frequency of the excitation is close to any of the natural frequencies of the structure.

To do a prestressed analysis, **Include geometric nonlinearity** must be selected in the **Frequency Domain** study step. This is automatic when you add the Frequency Domain, Prestressed study type.

The prestress loading can include a contact analysis, in which case the subsequent frequency domain analysis provides a linearization around the current contact state.

- Prestressed Structures
- Harmonic Perturbation

Bracket — Frequency-Response Analysis: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_frequency**.

OBTAINING A TIME HISTORY

Sometimes you want to study the time history over a period for the results of a frequency domain analysis. You can do that by adding a Frequency to Time FFT study step. The frequency response results are then viewed as terms in a Fourier series, which can be transformed to time domain. It is possible combine the results for several frequencies into a single time history, under the assumption that they are all multiples of the same fundamental frequency.

For examples showing how to obtain a time history from frequency domain results, see

- *Viscoelastic Structural Damper*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_frequency**.
- *Vibration Analysis of a Deep Beam*: Application Library path **Structural_Mechanics_Module/Verification_Examples/vibrating_deep_beam**.

Mode Superposition

Analyzing forced dynamic response for large models can be very time consuming. You can often improve the performance dramatically by using the mode superposition technique. The following requirements must be met for a modal solution to be possible:

- The analysis is linear. It is possible, however, that the structure has been subjected to a preceding nonlinear history. The modal response can then be viewed as a linear perturbation around that state.

- There are no nonzero prescribed displacements. For the common special case that all constrained points have the same synchronous displacement history, you can use a Base Excitation node to circumvent this limitation.
- The important frequency content of the load is limited to a range that is small when compared to all the eigenfrequencies of the model, so that its response can be approximated with a small number of eigenmodes. In practice, this excludes wave and shock type problems.
- If the modal solution is performed in the time domain, all loads must have the same dependency on the time. This requirement can be relaxed if you use a reduced-order model to represent the system, rather than using one of the mode superposition studies.

When using the Structural Mechanics Module, there are three predefined study types for mode superposition:

- Time Dependent, Modal
- Frequency Domain, Modal
- Frequency Domain, Prestressed, Modal

The two first of these study types consist of two study steps: One step for computing the eigenfrequencies and one step for the modal response. The last one has three study steps. Before the eigenfrequency step, you solve a static load case in order to get a prestress state used in the eigenfrequency computation.

In practice, you have often computed the eigenfrequencies already, and then want to use them in a mode superposition. In this case, start by adding an empty study, and then add a **Time Dependent, Modal** or **Frequency Domain, Modal** study step to it. After having added the study step this way, you must point the modal solver to the solution containing the eigenfrequencies and eigenmodes. You do this by first selecting **Show default solver** at the study level, and then selecting the eigenfrequency solution to be used in the **Eigenpairs** section of the generated modal solver.

In a mode superposition analysis, the deformation of the structure is represented by a linear combination of its eigenmodes. The amplitudes of these modes are the degrees of freedom of the reduced problem. You must select which eigenmodes to include in the analysis. This choice is usually based on a comparison between the eigenfrequencies of the structure and the frequency content of the load. As a rule of thumb, select eigenmodes up to approximately twice the highest frequency of the excitation.

In the mode superposition formulation in COMSOL Multiphysics, the full model is projected onto the subspace spanned by the eigenmodes. A problem having the

number of degrees equal to the number of included modes is then solved. Note that this differs from many implementations of mode superposition, where it often is assumed that the modal equations are totally decoupled.

An implication of this, is that it is not necessary to assume a certain structure of the damping matrix. Any type of damping which is allowable in the corresponding analysis of the full system can also be used in the modal based analysis.

For many common cases, the mode superposition analysis is not sensitive to whether the eigenmodes were computed using damping or not. The reason is that the eigenmodes of problems with Rayleigh damping and loss factor damping can be shown to be identical to those of the undamped problem, so that the projection to the subspace spanned by the eigenmodes is the same in both cases. For more general damping, it is however recommended that you suppress all contributions to the damping during the eigenfrequency step, and thus base the mode superposition on the solution to the undamped eigenfrequency problem.

FREQUENCY DOMAIN ANALYSIS

All loads are assumed to have a harmonic variation. This is a perturbation type analysis, so only loads having the **Harmonic perturbation** property selected are then included in the analysis.

TIME-DEPENDENT ANALYSIS

Only the factor of the load which is independent of time should be specified in the load features. The dependency on time is specified as **Load factor** under the **Advanced** section of the modal solver. This factor is then applied to all loads.

- Modal Solver and Studies and Solvers in the *COMSOL Multiphysics Reference Manual*
- Mechanical Damping and Losses

Harmonic Perturbation

Analyses in the frequency domain assume that the problem you study is linear, at least with respect to the response to the harmonic excitation. There may be other nonlinearities, such that the structure has responded nonlinearly to a previous loading. This preload could, for example, have caused a large rotations or prestress of a rubber membrane.

The concept of *harmonic perturbation* is in COMSOL Multiphysics used for distinguishing the linear harmonic analysis from a possible prestress analysis. The most important implication is that if a load has the **Harmonic Perturbation** selection, it is applied only in a study that is of the perturbation type. A load without this selection is, on the other hand, ignored in such a study. In this way, two sets of loads can be distinguished from each other. Technically speaking, the effect of marking a load as **Harmonic Perturbation** is that the `linper()` operator is applied to the value of the load.

For most load types, the use of **Harmonic Perturbation** is straightforward, but some cases need a more detailed discussion:

- A **Rigid Connector** can be assigned a **Harmonic Perturbation** subnode in which you can prescribe harmonic perturbation values to constrained degrees of freedom. If you have added **Applied Force** or **Applied Moment** nodes under a **Rigid Connector**, you can independently assign **Harmonic Perturbation** to these nodes, so that the loads are considered as being of the perturbation type.
- Even though initial stresses and strains are not usually considered as loads, you can assign **Harmonic Perturbation** also to the **Initial Stress and Strain** nodes.
- Rather assigning **Harmonic Perturbation** to a load, you can write the load value enclosed in the `linper()` operator. This is particularly useful when the feature that provide the loading does not have the **Harmonic Perturbation** option.

The default settings for the different structural mechanics study types in the frequency domain are summarized in Table 2-3.

TABLE 2-3: DEFAULT PERTURBATION SETTINGS FOR STRUCTURAL MECHANICS STUDY TYPES.

STUDY TYPE	STUDY STEPS	PERTURBATION
Frequency Domain	Frequency Domain	No
Frequency Domain, Prestressed	Stationary Frequency Domain, Perturbation	No Yes
Frequency Domain, Modal	Eigenfrequency Frequency Domain, Modal	N/A Yes
Frequency Domain, Prestressed, Modal	Stationary Eigenfrequency Frequency Domain, Modal	N/A N/A Yes

- With the default settings you cannot use the same set of loads for a **Frequency Domain** and a **Frequency Domain, Modal** study because only the latter responds to perturbation loads.
- You can change the behavior of a **Frequency Domain** study to be of the perturbation type by modifying the solver sequence. In the **General** section of the settings for the **Stationary Solver**, change **Linearity** to **Linear perturbation**.

A solver that does not have **Linearity** set to either **Linear perturbation** or **Linear** may respond to nonlinear effects. There are multiphysics problems where this is wanted because there may be a nonlinearity in another physics, even though the harmonic solution within structural mechanics is linear. But if there are nonlinearities within the structural mechanics parts of the model, you must be careful with the solver settings.

In a **Frequency-Domain, Perturbation** study step, that is when **Linearity** is set to **Linear perturbation**, geometric nonlinearity will be incorporated in the sense that there is a split between the material and spatial frames. This makes it possible to take for example stiffness from follower loads into account, and to use a contact solution as linearization point.

In the *COMSOL Multiphysics Reference Manual*:

- Frequency-Domain Perturbation Study Step
- Harmonic Perturbation — Exclusive and Contributing Nodes
- Built-In Operators (`linper()` operator)

Linearized Buckling Analysis

A linearized buckling analysis can be used to estimate the critical load at which a structure becomes unstable. This is a predefined study type that consists of two study steps: An initial step in which a unit load is applied to the structure, and a second step in which an eigenvalue problem is solved for the critical buckling load.

COMSOL reports a *critical load factor*, λ,which is the multiplier to the initial load at which the structure becomes unstable. The corresponding eigenmode is the shape of the structure in its buckled state.

The value of the initial load is immaterial, since a linear problem is solved. If the initial load actually was larger than the buckling load, then the critical value of λ is smaller than 1. It is also possible that the computed value of λ is negative. This signifies that a reversed load will give the critical case.

When performing a linearized buckling study, it is possible to discriminate between live and dead loads, where the former are the ones with respect to which the critical load factor is computed, and the latter are assumed to be constant. In this case, two different basic load cases need to be solved before the eigenvalue solution.

The computed buckling modes can be used to provide an initial imperfection for a subsequent nonlinear buckling analysis,

- For more details about how to model buckling, see Buckling Analysis.
- The numerical formulation is described in the section Linear Buckling in the *Structural Mechanics Theory* chapter.
- Settings for the solvers are described in Studies and Solvers, Linear Buckling, and Buckling Imperfection in the *COMSOL Multiphysics Reference Manual*.

- *Bracket — Linear Buckling Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_linear_buckling**
- *Buckling Analysis of a Truss Tower*: Application Library path **Structural_Mechanics_Module/Buckling_and_Wrinkling/truss_tower_buckling**

Bolt Pretension Study

The **Bolt Pretension** study step is a special case of a **Stationary** study step, where the special degrees of freedoms used for modeling prestressed bolts are solved for. In all other study types, these degrees of freedom are inactive. Typically, you include a **Bolt Pretension** study step as the first step in a study in order to simulate the state after the assembly of a bolted joint. You can then add any other types of study steps for computing the effects of the service loads.

In a **Bolt Pretension** study step, it is possible to tension the bolts sequentially, or to change the bolt prestress more than once, if an auxiliary sweep over some history parameter is used.

Modeling Pretensioned Bolts

- *Modeling of Pretensioned Bolts*: Application Library path **Structural_Mechanics_Module/Tutorials/bolt_pretension_tutorial**
- *Prestressed Bolts in a Tube Connection*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/tube_connection**

Random Vibration (PSD) Study

A random vibration study is used to analyze the response of a structure subjected to loads that randomly vary in time, while their statistical properties do not change with time.

The input to a random vibration analysis is given in terms of power spectral densities (PSD) and, in the case of several loading sources, the load cross-correlations.

The results of this type of analysis can only be interpreted in a statistical sense. Usually, the root mean square (RMS) of the result quantities is studied.

The analysis is based on a mode superposition and the reduced-order model (ROM) functionality. Except from the computation of eigenfrequencies and corresponding eigenmodes, and the creation of the reduced model, the core of the computation is performed during result evaluation.

The **Random Vibration (PSD)** study is mainly an entry point when adding studies. When you select it, you actually get two studies and a number of nodes under **Global Definitions** added to the model.

- See Performing a Random Vibration Analysis for a detailed description of how to study random vibration.
- The theory is described in Random Vibration Theory.

- *Bracket — Random Vibration Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_random_vibration**
- *Random Vibration Test of a Motherboard*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/motherboard_random_vibration**
- *Random Vibration Analysis of a Deep Beam*: Application Library path **Structural_Mechanics_Module/Verification_Examples/random_vibration_deep_beam**

Response Spectrum Analysis Study

Response spectrum analysis is used for computing an approximation of the structural response to transient nondeterministic events, such as earthquakes or shocks.

The **Response Spectrum Analysis** study is mainly an entry point when adding studies. What you actually get when you add such a study is an **Eigenfrequency** study step together with a **Response Spectrum** node under **Definitions**.

The actual response is computed on demand during result evaluation, using the computed eigenfrequencies and modes. The settings for the evaluation are done in the **Response Spectrum 2D** and **Response Spectrum 3D** datasets.

If your response spectrum evaluation requires inclusion of a missing mass correction, you need to also compute a set of stationary load cases. To set up that analysis, use the **Create missing mass correction study** button ( ) on the header of the **Response Spectrum** section in the **Response Spectrum** node settings.

- See Performing a Response Spectrum Analysis for a detailed description of how to work with response spectrum evaluations.
- The theory is described in Response Spectrum Analysis Theory.
- The settings for the special datasets are described in Response Spectrum 2D and Response Spectrum 3D in the *COMSOL Multiphysics Reference Manual*.
- The Response Spectrum node is described in the *COMSOL Multiphysics Reference Manual*.

- *Earthquake Analysis of a Building*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/building_response_spectrum**
- *Shock Response of a Motherboard*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/motherboard_shock_response**

Inertia Relief Study

The inertia relief study is a special combination of two stationary study steps through which you can study unconstrained structures where a set of external loads are balanced by inertial forces. The inertial forces and corresponding accelerations that are needed for this balance is a result of the analysis.

- See Performing an Inertia Relief Analysis for a detailed description of how to set up an inertia relief study.
- For the underlying theory, see Inertia Relief in the *Structural Mechanics Theory* chapter.

Bracket — Inertia Relief Analysis: Application Library path
Structural_Mechanics_Module/Tutorials/bracket_inertia_relief

Selecting the Physics Interface

The structural mechanics products contain a number of physics interfaces for a wide range of applications. This section contains some guidelines for how to select an appropriate physics interface for your analysis.

The basic physics interfaces for structural mechanics are:

- Solid Mechanics — General modeling of solid objects with extension in all directions
- Elastic Waves, Time Explicit — For efficient analysis of linear elastic wave propagation in solid domains
- Shell and Plate — For objects which are thin in one direction, but have significant bending stiffness
- Layered Shell — A special shell interface targeting multilayered shells. It is available with the Composite Materials Module.
- Membrane — For objects which are thin in one direction, and have negligible bending stiffness
- Beam — For objects where two directions have significantly smaller dimensions than the third; significant bending stiffness
- Beam Cross Section — A special type of interface used to compute cross section properties of general beam cross sections
- Pipe Mechanics — For analysis of pipes with internal pressure. This interface is similar to the Beam interface, but specialized for analysis of pipes.
- Truss — For objects where two directions have significantly smaller dimensions than the third; only axial forces can be transmitted
- Wire — Similar to Truss, but only tensile forces can be transmitted

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

Solid Mechanics

Solid Mechanics is the most general approach for modeling a structural mechanics problem and is based on general principles of continuum mechanics. There are several

physics interfaces based on the principles of solid mechanics, where the Solid Mechanics interface contains the largest number of material models, and the most advanced boundary conditions.

The drawback with using solid mechanics is that the models can become computationally expensive, especially in 3D. For structures which are thin or slender, you should consider using one of the specialized physics interfaces: Shell and Plate, Membrane, Beam, Pipe Mechanics, Truss, or Wire.

3D SOLID GEOMETRY

The degrees of freedom (dependent variables) in 3D are the global displacements u, v, and w in the global x, y, and z directions, respectively.

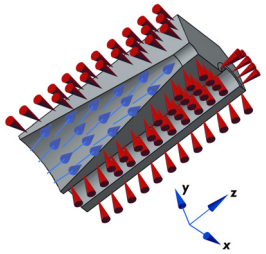


Figure 2-1: Loads and constraints applied to a 3D solid.

2D GEOMETRY

Plane Stress

Plane stress is useful for analyzing thin in-plane loaded plates. For a state of plane stress, the out-of-plane components of the stress tensor are zero.

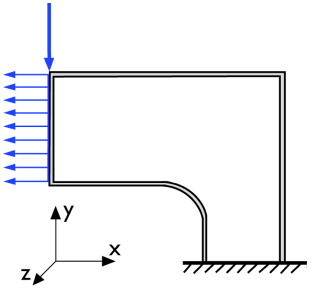


Figure 2-2: Plane stress is used to model plates where the loads are only in the plane; it does not include any out-of-plane stress components.

Plane stress allows loads in the x and y directions, and assumes that these are constant throughout the material's thickness, which can vary with x and y. The plane stress condition prevails in a thin (compared to the in-plane dimensions) flat plate in the xy-plane loaded only in its own plane and without any z direction restraint.

The plane stress conditions are enforced by adding extra degrees of freedom for the transverse strain, which are determined so that the transverse stress is zero in a weak sense.

Plane Strain

The plane strain assumption is that all out-of-plane strain components of the total strain tensor ε_z, ε_{yz}, and ε_{xz} are zero.

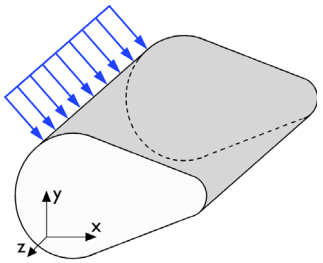


Figure 2-3: A geometry suitable for plane strain analysis.

Loads in the x and y directions are allowed. The loads are assumed to be constant throughout the thickness of the material, but the thickness can vary with x and y. Formally, the plane strain conditions require that the ends of the object are constrained in the z direction, but it is often also used for central parts of an object that is long in the z direction (compared to the in-plane dimensions). One example is a long tunnel along the z-axis where it is sufficient to study a unit-depth slice in the xy-plane.

Generalized Plane Strain

Generalized plane strain is similar to plane strain in the sense that transverse stresses can develop in the 2D cross section of a long object. The requirement that the out-of-plane strain is zero, is however relaxed. Instead, an assumption about zero resulting force in the transverse direction is used. Optionally, assumptions about zero bending moments over the cross section can be added. Generalized plane strain conditions

prevail in the inner parts of a long object with free ends. For many cases, generalized plane strain conditions is the 2D approximation that is closest to a full 3D solution.

For an in-depth discussion of different aspects of 2D solid mechanics, see www.comsol.com/blogs/what-is-the-difference-between-plane-stress-and-plane-strain

2D AXISYMMETRIC GEOMETRY

Solid mechanics in 2D axisymmetry uses cylindrical coordinates r, φ (`phi`), and z. All properties are independent of the azimuthal angle φ.

By default, displacements occur only in the rz-plane, and there are two degrees of freedom, u and w. By selecting the **Include circumferential displacement** option, you can model also torsion around the axis of rotational symmetry. The azimuthal rotation degree of freedom v is then included. In addition, many features such as loads, allow values to be entered in the φ direction.

The 2D axisymmetric geometry is viewed as the intersection between the original axially symmetric 3D solid and the half plane $\varphi = 0$, $r \geq 0$. Therefore, the geometry is drawn only in the half plane $r \geq 0$, and the original 3D solid is recovered by rotating the 2D geometry about the z-axis.

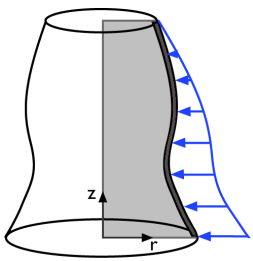


Figure 2-4: Rotating a 2D geometry to recover a 3D solid.

When working with resultants, for example loads and reaction forces, for an axisymmetric model, they refer to the whole circumference.

- *Axisymmetric Twist and Bending*: Application Library path **Structural_Mechanics_Module/Verification_Examples/axisymmetric_twist_and_bending**

1D GEOMETRY

Modeling solid mechanics in 1D implies that there are no variations in two transverse directions. For pure structural mechanics, this is a situation that is seldom of interest in practical engineering. The main application is to study various multiphysics phenomena.

The geometry is a line, and the degree of freedom (dependent variable) is the displacement u in the global x direction. Plane stress, plane strain, or generalized plane strain assumptions can be used. The out-of-plane formulation can be selected independently in the y and z directions.

For plane stress, additional degrees of freedom are added to account for the strains in the out-of-plane directions.

The generalized plane strain assumption adds one global variable for the additional strain in the y or z direction, and the corresponding transverse strain component can vary linearly throughout the cross-sectional area.

1D AXISYMMETRIC GEOMETRY

Solid mechanics 1D axisymmetry uses cylindrical coordinates. All properties are independent of the azimuthal angle φ and the axial z direction.

The geometry is a line, which represents a disk because of the axial symmetry. The line is drawn only in the half plane $r \geq 0$. Displacement occurs only in the radial direction, and there is one degree of freedom, u, the displacement in the global r direction.

Plane stress, plane strain, or generalized plane strain assumptions can be used in the axial direction. For plane stress, one additional degree of freedom is added to account for the strain in the z direction. The generalized plane strain assumption adds one global variable for the additional strain in the z direction.

Elastic Waves, Time Explicit

The Elastic Waves, Time Explicit interface is used to compute the velocity and strain field in solids with propagating linear elastic waves. In the Add Physics dialog, it is found in the **Acoustics** branch, under **Elastic Waves**.

In general, the interface is suited for modeling the propagation of waves over large distances relative to the wavelength, for example, ultrasound propagation for nondestructive testing (NDT), or seismic waves. The interface exists in 2D (generalized plane strain) and 3D.

The interface is based on the discontinuous Galerkin (dG or dG-FEM) method and uses a time explicit solver. The method is very memory efficient and can solve problems with many million degrees of freedom (DOFs). The method is also well suited for distributed computing on clusters.

In the *Acoustics Module User's Guide*:

- The Elastic Waves, Time Explicit Interface
- Theory for the Elastic Waves, Time Explicit Interface

Shell and Plate

The Shell interface is useful when the object is thin in one direction. Structures built from welded or bolted flat plates are archetypal shell structures and so are pressure vessels. The Shell interface is available in 3D and 2D axisymmetry.

The Plate interface is a specialization of the Shell interface, used for 2D modeling in the *XY*-plane. A plate model has its main action in bending out of the plane, but can optionally also treat in-plane forces. If the loads act only in the plane, using Solid Mechanics with the plane stress option is a better choice.

Shells are modeled on boundaries, and the kinematics in the transverse direction is represented only by the mathematical model. The degrees of freedom consist of displacements and rotations at the modeled boundary. This results in an assumption where the in-plane strains vary linearly through the thickness, and the stress in the thickness direction is zero. The thickness of a shell does not have to be constant, although this is by far the most common case.

When the Composite Materials Module is available, it is possible to model also multilayered shells.

For nonlinear material models, a layered approach with a single layer is used. There is a virtual mesh in the thickness direction, in order to accommodate the potential variation of the material properties in the thickness direction.

Rather than computing the shell stiffness from material properties and thickness, you can also directly enter that stiffness properties in tension, bending, and shear.

The Shell and Plate interfaces can be used both for "thin" and "thick" shells. Shear deformations are taken into account; this is usually called Mindlin theory.

When modeling with shells, it is important to keep track of "top" and "bottom" side when applying loads and interpreting the results.

The in-plane stiffness of an elastic shell is proportional to the thickness h, while the bending stiffness is proportional to h^3. The difference in stiffness along different directions can thus become very large. When an object is very thin, a shell model may be numerically ill-posed due to the negligible bending stiffness. It is then better to use the Membrane interface.

For more information, see Modeling with Plates and Shells.

Membrane

The Membrane interface can be used for very thin objects, like cloth, where only in-plane forces are important. Membranes can be considered as plane stress elements but with an arbitrary, possibly curved, configuration in space. The Membrane interface is available in 3D and 2D axisymmetry.

In most applications, a membrane must be pretensioned in order to have a stable configuration, so it will almost invariably be used in a geometrically nonlinear analysis. The main exception is when it is used as a "cladding" on top of a solid, since it will then be stabilized by the solid.

You can study configurations when there is local wrinkling in a membrane by adding a special nonlinear material model.

In the Membrane interface a large number of different material models can be used. When the Composite Materials Module is available, it is possible to model also multilayered membranes.

For more information, see Modeling with Membrane Elements.

Beam

A beam is an abstract model where only the extension in the axial direction is modeled explicitly on an edge. The cross section is usually specified in terms of geometrical

properties such as area and moments of inertia. Several predefined cross-section types are also available. The Beam interface is available in 3D and 2D.

The exact stress distribution in the beam is not explicitly modeled. It is actually not even fully determined by only the cross-sectional properties. Instead, six (in 3D) resultant section forces are used: axial force, shear forces in two perpendicular directions, two bending moments, and one twisting moment.

The Beam interface assumes linear elasticity.

Two formulations are available in the Beam interface:

- The classical Euler–Bernoulli beam theory, which is applicable for slender beams.
- Timoshenko theory, where also shear deformations are considered. This theory makes it possible to use the Beam interface to model rather thick beams.

For more information, see Modeling with Beam Elements.

Beam Cross Section

The Beam Cross Section interface is used to compute cross-section properties of beams, and also to evaluate stresses in a beam segment when the section forces along the beam are known. It is mainly used in conjunction with the Beam and Pipe Mechanics interfaces.

As input, you provide a 2D sketch of the cross section. In addition to the stress variables, geometrical properties like area, moments of inertia, center of gravity, principal axis directions, torsional rigidity, shear center location, shear correction factors, and warping constant are calculated.

The Beam Cross Section interface is available in 3D and 2D. The full 3D stress distribution along the beam can be visualized.

- *Beam Section Calculator*: Application Library path **Structural_Mechanics_Module/Applications/beam_section_calculator**

Pipe Mechanics

Pipes are similar to beams, and many properties of the Pipe Mechanics interface are shared with the Beam interface. The most distinguishing feature is that the internal pressure usually causes a significant part of the stresses in a pipe. Also, temperature gradients usually occur through the pipe wall, rather than across the entire section.

The loads from internal pressure and drag forces can be taken directly from results in the Pipe Flow interface. Similarly, the temperature in the pipe walls can be taken from the Heat Transfer in Pipes interface.

In the *Pipe Flow Module User's Guide*:

- The Pipe Flow Interface
- The Heat Transfer in Pipes Interface

Truss

The truss interfaces have two main purposes:

- Modeling of trusses, consisting of straight bars carrying only axial forces
- As reinforcements, used in conjunction with other physics interfaces

The truss interfaces are available in 3D and 2D.

For a truss model, only one geometrical property is needed, the cross-section area.

A truss element has no stiffness in the directions perpendicular to its extension. For trusses, this is usually not a problem since they are designed such that each member is stabilized by its neighbors.

For more information, see Modeling with Truss Elements.

Wire

The Wire interface intended for modeling of cables and wires. The structures are often prestressed, but also cables sagging under, for example, self-weight can be modeled.

The main difference between the Truss interface and the Wire interface is that the wires cannot sustain any compressive forces. In reality the wire will wrinkle. Numerically, this is handled by using a very low stiffness in compression.

For more information, see Modeling Wires and Cables.

Selecting Discretization

Shape Function Order

In structural mechanics analysis, the focus is often on the stresses and strains rather than on the displacements. Since the strains are derivatives of the displacement field, the accuracy of the strains will be one order less than the accuracy of the displacements. For this reason, second-order shape functions are used as default in most of the structural mechanics interfaces. Often this gives the best trade-off between model size and accuracy.

It is well known that using first-order shape functions in solid mechanics will give an overly stiff solution, unless a very fine mesh is used. This is especially noticeable for triangular and tetrahedral elements. This can, to some extent, be counteracted by using reduced integration, see Using Reduced Integration.

If the purpose of the analysis is only to compute stiffness, rather than stresses, the use of linear shape functions can still be justified. This is the default choice in the Multibody Dynamics interface, available with the Multibody Dynamics Module.

If the solution contains discontinuities, for example when some type of front is moving through the material, first-order elements and a fine mesh is often a good choice, since the advantage of the higher-order elements lies in their ability to represent smooth gradients.

TRUSS ELEMENTS

In the Truss interface, the default is to use first-order shape functions, since the elements are mainly used in a context where the axial force in each element is constant.

When truss elements share an edge with other structural elements, you should choose the same discretization in both interfaces, usually quadratic.

BEAM ELEMENTS

The beam elements have only one set of shape functions, which cannot be changed. The axial displacement and the twist are represented by first-order shape functions, while the bending is represented by cubic Hermitian shape functions. This element can then represent a constant axial force, a constant twisting torque, a linear bending moment, and a constant shear force. This is the exact solution for a beam having no distributed loads.

A consequence of this formulation is that it may not possible to obtain a perfectly conforming approximation if a beam shares an edge with elements from another physics interface.

Lagrange and Serendipity Shape Functions

In the Solid Mechanics and Membrane interfaces, you can choose between two families of shape functions: *Lagrange* and *serendipity*. The default is to use serendipity shape functions.

The serendipity elements have the advantage of generating significantly fewer degrees of freedom for structured meshes. The accuracy is in most cases almost as good as for the Lagrange elements. The Lagrange elements are however less sensitive to strong mesh distortions.

The serendipity shape functions differ from the Lagrange shape functions only for the following element shapes:

- 2D: Quadrilateral elements of discretization order higher than 1
- 3D: Hexahedral, prism, and pyramid elements of discretization order higher than 1

In the *COMSOL Multiphysics Reference Manual*:

- The Lagrange Element (shlag)
- The Nodal Serendipity Element (shnserp)

When coupling two structural mechanics physics interfaces, the same type of shape functions should be used in both interfaces to ensure conformity in displacement shape functions. Since there is no difference between the two families of shape functions in 1D, this is not an issue when connecting edges.

Choosing Shape Functions in Multiphysics Models

In problems where several physics fields participate, the accuracy can sometimes be improved by considering how the different fields interact. In structural mechanics, it is common that other physics fields directly affect the inelastic strains. This is the case in, for example, thermal expansion and hygroscopic swelling.

In thermal expansion, the elastic strain used in most constitutive relations is the difference between the total strain, which is computed from derivatives of the displacement field, and the thermal strain:

$$\varepsilon_{el} = \varepsilon_{tot} - \varepsilon_{th} = \varepsilon_{tot} - \alpha(T - T_{ref})$$

Since the thermal strain is directly proportional to the temperature, a consistent approximation would be to use one step lower discretization order for the temperature than for the displacements. When using built-in multiphysics couplings, such as **Thermal Expansion**, such a modification is not necessary. Any mismatch is automatically taken care of by reinterpolating the thermal strains to an order that matches the general strain field.

Another type of coupling appears on the boundary between two domains having different physics, as in fluid–structure interaction and acoustic–structure interaction. When, for example, Thermoviscous Acoustics is coupled to Solid Mechanics, then the time derivative of the displacement in the solid is set equal to the velocity in the acoustic medium on the shared boundary. In this case, it makes sense to have the same shape function order for these two fields.

For bidirectional multiphysics couplings, it is not always possible to select shape function orders for all fields that will give perfectly consistent representation of all terms.

Implicit Shape Function Orders

Some solid mechanics formulations contain other degrees of freedom in addition to the displacements. The shape functions are then selected internally based on your choice of displacement discretization.

VISCOELASTIC STRESS AND STRAIN

When using a viscoelastic material model, auxiliary degrees of freedom are added either for the viscoelastic strains or the viscoelastic stresses, depending on whether a linear or a nonlinear formulation is used. These degrees of freedom are local to the element, and you can select either the discontinuous Lagrange or Gauss-point data type.

The discontinuous Lagrange shape functions will have an order that is one below what is used for the displacement shape functions.

If Gauss-point data is used, the same integration points as used for the numerical integration of the stiffness matrix are used. This order depends on the selected displacement discretization order and whether reduced integration is used or not.

INELASTIC STRAINS

For material models like plasticity and creep, the inelastic strains are formally degrees of freedom. They will be allocated at the same integration points that are used for the numerical integration of the stiffness matrix. This order depends on the selected displacement discretization order and whether reduced integration is used or not.

Using Reduced Integration

Using reduced integration in structural mechanics often has two main benefits:

- It reduces the computational effort per mesh element for assembly of system matrices.
- It improves the accuracy of the finite element by, for example, alleviating locking and an overly stiff behavior.

The reduced computational effort stems from the fewer quadrature points in which expressions are evaluated during the assembly of the system matrices. This reduction can be particularly beneficial for models where a comparably large time is spent in assembly, as compared to the time needed to solve the system of equations. Typical examples include structural dynamics, where the many increments can lead to a significant time spent on assembling, and nonlinear material models where each assembly step can be expensive. Plasticity, for example, involves solving a system of nonlinear equations at each quadrature point, thus reducing the number of quadrature points can significantly speed up the assembly time. For material models that add auxiliary dependent variables, for example viscoelasticity, using reduced integration will also reduce the amount of memory required to solve the model.

Reduced integration can be used selectively for different material models within the same physics interface.

DISPLACEMENT ORDER AND MESH ELEMENT TYPES

The accuracy and performance of finite elements derived using a reduced integration scheme depends on both the shape order of the displacement field and on the mesh element type. For some combinations, reduced integration will give "exact" results, whereas other combinations may result in deformation modes that produce zero strain energy — so-called *spurious zero energy modes* or *hourglass modes*. In the latter case, stabilization is needed to suppress such unwanted modes of deformation.

Linear

Using a linear displacement shape order is typically not encouraged in structural mechanics, since all mesh element types lead to a deficient formulation. The derived finite elements can be considered overly stiff, especially when a coarse mesh is used. This overly stiff behavior is commonly caused by shear-locking effects.

These deficiencies can be alleviated by using a reduced integration scheme, which, for a linear shape order, results in a quadrature order equal to zero. Having only a single

integration point per mesh element removes the shear locking for linear elements, and it can significantly improve the accuracy in coarse meshes. However, the derived finite elements exhibit severe hourglassing and require stabilization to be useful. An example of a two-dimensional bending problem is given in Figure 2-5, which shows the performance of linear quadrilaterals using reduced integration with and without stabilization. A so-called hourglass pattern emerges without stabilization making the solution clearly not viable.

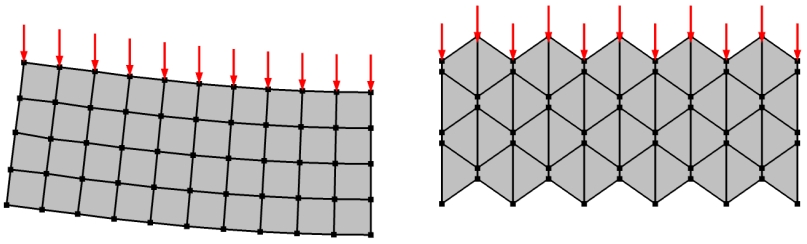


Figure 2-5: Two-dimensional bending example using a linear displacement field and reduced integration with stabilization (left) and without it (right).

Using reduced integration for a linear displacement shape order can also remove volumetric locking and improve the accuracy for incompressible materials, either with or without a mixed formulation.

The derived finite elements are still sensitive to hourglassing even with stabilization, and singularities often excite even stabilized hourglass modes. It is therefore recommended to avoid features that implement

- Point and edge loads
- Point and edge constraints

when using a linear displacement shape order.

Linear triangular and tetrahedral mesh elements exhibit poor accuracy also when using reduced integration, and are generally not recommended.

For quadrilateral and hexahedron mesh elements, the accuracy of the finite element and the efficiency of the hourglass stabilization is best for mesh element with a good aspect ratio and quality. The same applies to prism mesh elements.

Quadratic Serendipity

Finite elements derived using quadratic serendipity shape functions and reduced integration in general have a less stiff behavior than their fully integrated counterparts.

The coarse mesh accuracy in terms of displacements can therefore be improved by using reduced integration.

If the displacement field uses quadratic serendipity shape functions, any mesh element leads to a finite element that has no hourglass modes, and therefore no stabilization is needed when using reduced integration. For this reason, using quadratic serendipity shape functions together with reduced integration can be a very efficient formulation. In theory, there could appear hourglass modes, but these are suppressed when the finite elements are assembled.

Stabilization is never added when serendipity shape functions are used for the displacement field.

Quadratic Lagrange

Finite elements derived using quadratic Lagrange shape functions and reduced integration in general have a less stiff behavior than their fully integrated counterpart. The coarse mesh accuracy in terms of displacements can therefore be improved by using reduced integration.

However, when using reduced integration some mesh element types may lead to finite elements that exhibit hourglass modes, this is the case for quadrilaterals and hexahedra. Although these finite elements are not as sensitive as their linear counterpart, they require stabilization to be viable. Stabilization is also encouraged for mesh elements that have deformation modes with a very small stiffness, such as pyramids and prisms.

Higher-Order Shape Functions

Higher-order shape functions share many characteristics with the corresponding quadratic shape functions. That is, there is no need for stabilization when using serendipity shape functions, whereas finite elements derived from Lagrange shape functions will have hourglass modes that require stabilization.

Triangular and Tetrahedral Mesh Elements

Triangular and tetrahedral mesh elements are special cases for which the reduced integration scheme is exact given that the corresponding mesh element is not distorted. These mesh element types do not exhibit hourglass modes when using reduced integration, and therefore stabilization is not needed.

However, it is not recommended to use triangular or tetrahedral mesh elements in combination with a linear displacement shape order.

HOURGLASS STABILIZATION

A number of energy quantities are added that can be used to evaluate the amount of hourglass stabilization energy introduced, and also to portrait parts of the model that are prone to hourglassing. These include:

- `<phys>.Wstb`, a field showing the stabilization energy density. Its interpretation depends on the stabilization method used.
- `<phys>.Wstbavg`, a mesh element average of the variable `<phys>.Wstb`. This variable is not available for layered shell features.
- `<phys>.Wstb_tot`, the total amount of stabilization energy in the model.

As a crude rule-of-thumb, the total stabilization energy should not be more than 5% of the total strain energy in the model. If this is not the case, revise the model or use full integration.

Always make sure that the obtained solution does not exhibit hourglassing when using reduced integration with formulations prone to it. This is especially the case when the displacement field is linear, since such finite elements can be particularly sensitive to hourglassing.

Energy Sampling, Hessian and Flanagan–Belytschko

The total hourglass stiffness k_{stb} used by the Energy Sampling, Hessian and Flanagan–Belytschko stabilization methods is schematically determined like

$$k_{\text{stb}} = \max(f_{\text{stb}} \cdot f_{\text{in}}, f_{\text{min}}) k_{\text{stb0}}$$

Here f_{stb}, f_{in}, and f_{min} are dimensionless stiffness multipliers used to scale the base estimate of the hourglass stiffness given by k_{stb0}. The value of k_{stb0} depends on the stabilization method but in general depends on material properties, mesh element type, and the quality of the mesh element. The three multipliers have the following meaning

- The multiplier f_{stb} is a user-defined expression of any parameter or variable in the model. It defaults to 1.
- The multiplier f_{in} is internal and used to account for various inelastic deformations like plasticity and damage.
- The multiplier f_{min} is used set a lower limit for the hourglass stiffness.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter.

Modeling with Plates and Shells

In this section:

- Introduction
- Dependent Variables
- Layered and Nonlayered Shells
- Working with Section Stiffness
- Loads

Introduction

Shells and plates are structures which are significantly thinner in one direction than in the other two directions. The mathematical description can then be simplified by using the following assumptions:

- The deformation field is described by the displacements of the midsurface plus the rotation of the midsurface normal.
- The in-plane total strains vary linearly through the thickness.
- The stress in the transverse direction is negligible.

These assumptions make it possible to reduce the number of degrees of freedom significantly from what would be needed for a corresponding 3D solid model. In addition, there are advantages in terms of numerical conditioning and meshing.

In shell theory, all quantities refer to the *midsurface*. In the Shell interface, you can however use any boundary that is parallel to the midsurface for modeling. If the *reference surface* where the mesh is located is not the midsurface, you can use an offset distance to describe the true location. If you have a full 3D geometry, there is thus no need to create the midsurface. You can place the mesh on either the top or bottom surface of the actual geometry.

Plates are similar to shells but act in a single plane and usually with only out-of-plane loads. The plate and shell elements in COMSOL Multiphysics are based on the same formulation. The Plate interface for 2D models is a specialization of the Shell interface. In the following, the text fully describes the Shell interface, and the Plate interface is mentioned only where there are nontrivial differences.

A Shell interface can be active on:

- Free surfaces embedded in 3D.
- The boundary of a solid 3D object. In this case, it can be used to model a reinforcement on the surface of a 3D solid.
- Free boundaries in 2D axisymmetry.
- The boundary of a solid object modeled in 2D axisymmetry. In this case, it can be used to model a reinforcement on the surface of a 3D solid.

A Plate interface can only be active on domains in 2D.

Dependent Variables

The element used for the shell interface is of Mindlin-Reissner type, which means that transverse shear deformation is accounted for. It can thus also be used for rather thick shells. In 3D and for plates, an MITC (*mixed interpolation of tensorial components*) formulation is used. A general description of this element family is given in Ref. 167.

The dependent variables in 3D are the displacements u, v, and w in the global x, y, and z directions, and the displacements of the shell normals a_x, a_y, and a_z in the global x, y, and z directions.

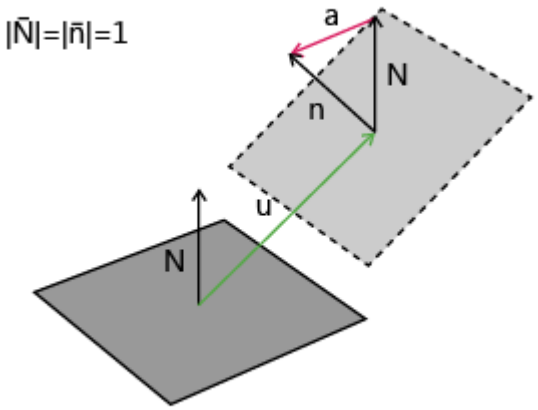


Figure 2-6: The degrees of freedom in the shell interface. **N** *is the normal vector in the original configuration and* **n** *is the normal in the deformed state.*

The degrees of freedom represent the displacements on the reference surface. If an offset property is used, the reference surface differs from the physical shell midsurface. The displacement vector on the midsurface, **u**, can be expressed as

$$\mathbf{u} = \mathbf{u}_R + \zeta_0 \mathbf{a}$$

where $\mathbf{u}_R$ is the displacement on the reference surface (the displacement degrees of freedom) and ζ_0 is the offset. The rotational displacement **a** is the same on both midsurface and reference surface.

In axisymmetry, there are four degrees of freedom, since **u** and **a** only have components in the *RZ*-plane.

For input and output, the Shell interface to a large extent replaces the displacements of the shell normals by the more customary rotations θ_x, θ_y, and θ_z about the global axes. For a geometrically linear analysis, the relation between normal displacement and rotation vector is simple: $\mathbf{a} = \theta \times \mathbf{n}$ where $\mathbf{n}$ is the unit normal of the shell.

For a standard plate analysis, only three degrees of freedom are needed: the out-of-plane displacement w and the displacements of the shell normals α_x and α_y. It is also possible to activate all six degrees of freedom, so that any type of analysis of a shell initially positioned in the xy-plane can be performed using the Plate interface. Using six degrees of freedom is the default, but three degrees of freedom can be selected instead for efficiency.

For plates, the rotations θ_x, θ_y, and (possibly) θ_z are used to a large extent.

- When six degrees of freedoms are used in the Plate interface, there must be enough constraints to suppress any in-plane rigid body motions.
- In the Shell interface, the coordinates are usually denoted with lowercase letters (x, y, z). If a Solid Mechanics or Membrane interface is present in the same model, then it becomes necessary to make a difference between the material frame and the spatial frame (Material and Spatial Coordinates). In this case, the coordinates in the Shell interface changes to (X, Y, Z).

Layered and Nonlayered Shells

The Shell interface has two fundamental types of material models. The first is represented by Linear Elastic Material, Rigid Material, and Section Stiffness. The other type of material models consists of Linear Elastic Material, Layered, Hyperelastic Material, Layered and Piezoelectric Material, Layered. In either case, the dependent variables are the same, and exist only on the reference surface. The fundamental difference is that in the first group, the material properties are assumed to be constant through the thickness, so that stiffness and mass matrices can be computed by an analytical integration in the thickness direction.

In the **Linear Elastic Material, Layered**, **Hyperelastic Material, Layered** and **Piezoelectric Material, Layered** models, there is a numerical integration in the thickness direction. It is also possible to store states, such as inelastic strains, at different through-thickness locations. Thus, **Linear Elastic Material, Layered** forms the basis for all nonlinear material

models even if the shell is not layered as such. Use this material model if you want to write your own expressions as function of through-thickness location.

When the Nonlinear Structural Materials Module is available, **Linear Elastic Material, Layered** can be used to model plasticity, creep, and other nonlinear materials; and when the Composite Materials Module is available it can be used to model multilayered shells.

Linear Elastic Material, Layered, **Hyperelastic Material, Layered**, and **Piezoelectric Material, Layered** models are only available for the Shell interface, not for the Plate interface.

The accuracy of the results in a layered material model depends on the resolution in the thickness direction. For each layer, you have the option to set the resolution. In a layered material, this is the **Mesh elements** property in the layer definitions. When working with a single layer material, then it is the **Mesh elements** property in the **Shell** property group.

As this setting indicates, there is a virtual mesh in the transverse direction (the e*xtra dimension*). When there is a significant variation in the thickness direction, as is the case for plastic strains in state of bending, you need a good enough resolution.

The virtual mesh depends on the in-plane discretization, and so does the number of integration points in the thickness direction. In Table 2-4, the number of integration points that are used in the thickness direction are summarized.

TABLE 2-4: NUMBER OF INTEGRATION POINTS IN THICKNESS DIRECTION.

DISCRETIZATION	MESH ELEMENTS =1	MESH ELEMENTS = 2	MESH ELEMENTS = 3	MESH ELEMENTS = 4
Linear	2	4	6	8
Quadratic	3	6	9	12

For an example showing how to model an elastoplastic shell, see *Twisting and Bending of a Metal Frame*: Application Library path **Nonlinear_Structural_Materials_Module/Plasticity/frame_with_cutout_plasticity**.

Working with Section Stiffness

In some cases, shell models are used as a homogenized approximation of a thin inhomogeneous structure. Examples of this are perforated sheets and corrugated sheets, but also laminates. For such cases, a single material model together with a thickness does not contain enough information to set up the model.

The Section Stiffness material model provides an option to describe the shell properties directly in terms of quantities such as membrane stiffness, bending stiffness, and so on.

Actually, **Linear Elastic Material, Layered** is a special case of such a homogenized model, where the homogenization is performed in the thickness direction. The ABD matrices computed as side effect by this material model provides exactly the same form of data that is used as input in the **Section Stiffness** node.

Loads

Loads can be applied at an arbitrary location in the through-thickness direction. Placing loads on the correct side of the shell is particularly important in two cases:

- When the loads act in the plane of the shell, since a tangential traction on the top or bottom surface also creates a moment around the midsurface.
- When the shell is thick and curved, since the loaded area will then differ significantly between the two sides.

By using the settings in the **Through-Thickness Location** section present in various load features, you can take these effects into account.

Modeling with Layered Shells

A layered shell, also known as a composite laminate, is a thin-walled structure in 3D having many layers of different orthotropic (or optionally isotropic/anisotropic) material stacked on top of each other. A layered shell can be active either on free surfaces embedded in 3D or on the boundary of a solid object in 3D. In the latter case, it can be used to model a reinforcement on the surface of a solid object.

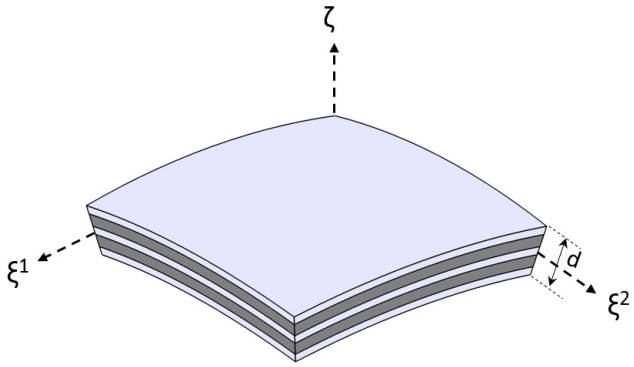


Figure 2-7: Geometry of a doubly curved laminated shell.

Figure 2-7 shows the uniform thickness doubly curved laminated shell having an orthogonal curvilinear coordinate system (ξ^1, ξ^2, ζ) and a total thickness (d).

A typical stacking sequence of a composite laminate having n layers is shown in Figure 2-8. The thickness of each layer (d_k), as well as the fiber direction in each layer (θ_k) with respect to first principal direction (ξ^1) of the laminate are indicated. A

counterclockwise rotation of the fiber direction with respect to the (ζ) direction is considered as positive.

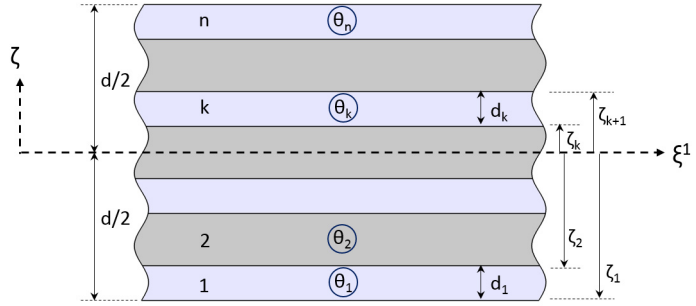


Figure 2-8: A typical stacking sequence of a composite laminate showing thickness and fiber orientation in each layer.

For a general description of this section, see About Laminated Composite Shells in the documentation for the Composite Materials Module.

In COMSOL Multiphysics, layered shells can be analyzed either by the layerwise theory using the Layered Shell interface, or by the first order shear deformation theory (ESL-FSDT) theory using the **Linear Elastic Material, Layered** material model in the Shell interface. The layerwise theory of modeling a layered shell is discussed in the next section.

The first order shear deformation theory (ESL-FSDT) for modeling a layered shell is described in Theory for FSDT Laminated Shell.

PHYSICS INTERFACE SELECTION

A Layered Shell physics interface is defined using a surface (2D) geometry and an extra dimension (1D) geometry in the through-thickness (or normal) direction. The surface geometry is a physical geometry and supposed to be created in the model whereas the extra dimension geometry is a virtual geometry created by **Layered Material** and similar nodes.

The geometric surface (or its lower dimension) selection, together with layer (or interface) selection creates the product geometry for the governing equations and boundary conditions of the Layered Shell interface.

The Layered Shell interface itself exists on a domain level which is a product geometry created using selected geometric surfaces and material layers. The physics features can exist on domain level or its lower dimensions. Based on whether a physics feature exists on a layer or an interface, the physics features can be divided into two categories:

- Layer features (having boundary, edge, or point selection)
- Interface features (having boundary, edge, or point selection)

Layer features may or may not allow a layer selection. When a layer selection is not allowed, the layer feature is applied to all layers.

Interface features, may allow only exterior interfaces, interior interfaces, or alternatively all interfaces of a laminate.

- For a general description of layer selections, see Layer and Interface Selections in the documentation for the Composite Materials Module.
- For a general description of layered materials, see Layered Materials in the documentation for the Composite Materials Module.

LAYER MATERIALS, THICKNESSES, AND ORIENTATIONS

A layered shell can have many layers with following different properties in each layer:

- Material (isotropic, orthotropic, or anisotropic)
- Thickness
- Orientation
- Mesh elements in the through-thickness direction

For a general description of layered materials, see Layered Materials in the documentation for the Composite Materials Module.

OFFSET AND LOCAL COORDINATE SYSTEM

The position of the reference surface with respect to midplane of the laminate and the local coordinate system in which material properties and results are interpreted can be defined in **Layered Material** and similar nodes.

For a general description of layered materials, see Layered Materials in the documentation for the Composite Materials Module.

TRANSFORM AND SCALE

The transform functionality can be optionally used to simplify the definition of stacking sequence. Various transform options available are symmetric, antisymmetric, and repeated.

The scale functionality can be optionally used to model a variable thickness layer. The scale defined for each layer, which could be a function of reference geometry, is multiplied to the constant layer thickness essentially making it a variable thickness layer.

The transform and scale options can be defined in **Layered Material** and similar nodes.

For a general description of layered materials, see Layered Materials in the documentation for the Composite Materials Module.

USING THE EXTRA DIMENSION COORDINATES

Sometimes, you want to write expressions that are functions of the coordinates in the thickness direction of the layered shell. If you write expressions based on the usual coordinates, like X, Y, and Z, such an expression will be evaluated on the reference surface (the meshed boundaries). In addition to this, you can access locations in the through-thickness direction by making explicit or implicit use of the coordinates in the extra dimension.

The extra dimension coordinate has a name like `x_llmat1_xdim`. The middle part of the coordinate name is derived from the tag of the layered material definition where it is created; in this example a **Layered Material Link**.

You can also access the extra dimension coordinate as wrapped into a physics interface variable, like `lshell.xd` (varies from 0 to the total laminate thickness d) and `lshell.xd_rel` (varies from 0 to 1).

Finally, the coordinates in 3D space are available using the physics scoped variables `lshell.X`, `lshell.Y`, and `lshell.Z`. These coordinates vary also in the thickness direction of the layered shell.

You can also write expressions that explicitly contain the number of the layer, available in the variable `lshell.num`. The number of the bottommost layer is '1'.

RESULTS EVALUATION IN LAYERED SHELLS

For visualization and results evaluation, predefined variables include all nonzero stress and strain tensor components, principal stresses and principal strains, in-plane and out-of-plane forces, moments, and von Mises and Tresca equivalent stresses.

Stresses and strains are available in the global coordinate system, laminate coordinate system, as well as in the layer local coordinate system.

Layered Material Slice Plot

It is possible to evaluate the stress and strain tensor components and equivalent stresses in each layer of a laminate using **Layered Material Slice** plot.

The through-thickness location can be set to evaluate a quantity in the middle of a layer, at an interface between two layers, top or bottom of a laminate, and so on. The top, bottom, and the middle of a laminate can be defined as

- Bottom of a laminate: -1 (relative) and 0 (physical)
- Middle of a laminate: 0 (relative) and $d/2$ (physical)
- Top of a laminate: $+1$ (relative) and d (physical)

The default value for the through-thickness location is given in the **Default through-thickness result location** section of the Layered Shell interface.

Through Thickness Plot

The through-thickness variation of a quantity at one or more locations on the reference surface can be plotted using a **Through Thickness** plot. In this plot, the reference surface locations can be specified through following ways:

- By selecting one or more geometric points
- By specifying the coordinate of one or more points
- By creating a cut point dataset

Layered Material Dataset

The **Layered Material** dataset allows the display of results in 3D solid even though the equations are solved on a 2D surface.

Using this dataset, results can either be visualized on a 3D object or on slices created in the through-thickness direction of a 3D object. The following options are available in the dataset to create slices in the through-thickness direction:

- Mesh nodes
- Interfaces
- Layer midplanes

Sometimes, when a laminate is very thin, it becomes difficult to distinguish between a surface or a solid object. In such cases it is possible to scale the through-thickness direction in the dataset for better visualization.

In order to use the section stiffness approach, you must obtain all terms in the stiffness matrix. In the case of dynamic analysis, also inertial matrices are needed. For some common cases, like perforation and corrugation, expressions for these quantities may be found in the literature. In general, however, you have to set up a finite element model of a representative part of the structure, and compute the homogenized data based on the results from various load cases.

Given only the homogenized stiffness and computed deformations, there is not enough information to compute stresses. The stress state will depend on the local geometry. In the **Section Stiffness** node, you have a possibility to define expressions for the relation between section forces and stresses. Such expressions must then be based on your knowledge of the peak stresses in the original, not homogenized structure. As an example, for a perforated structure, the stress concentrations around the holes should be taken into account.

Modeling with Membrane Elements

About Membranes

Membranes can be considered as plane stress elements in 3D with a possibility to deform both in the in-plane and out-of-plane directions. The difference between a shell and a membrane is that the membrane does not have any bending stiffness. If the ratio between the thickness and the dimensions in the other directions becomes very small, a membrane formulation is numerically better posed than a shell formulation.

The Membrane interface supports the same study types as the Solid Mechanics interface except it does not include the Linear Buckling study type, but wrinkling can be modeled instead.

To describe a membrane, provide its thickness and the material properties. All properties can be variable over the element. All elemental quantities are integrated only at the midsurface. This is a good approximation since by definition a membrane is thin.

The physics interface is intended to model either prestressed membranes or a thin cladding on top of a solid.

Membranes for 3D Models

The Membrane interface in 3D can be active on interior and exterior boundaries of a domain, as well as on boundaries not adjacent to any domain.

The dependent variables are the displacements u, v, and w in the global x, y, and z directions, and the displacement derivative u_{nn} in the direction normal to the membrane. For anisotropic materials, the tangential displacements derivatives $u_{1\mathrm{n}}$ and $u_{2\mathrm{n}}$ are additionally added as dependent variables.

Membranes for 2D Axisymmetric Models

The Membrane interface for 2D axisymmetric models can be active on interior and exterior boundaries of a solid, as well as on edges that not adjacent to a solid.

The dependent variables are the displacements u and w in the global r and z directions, and the displacement derivative u_{nn} in the direction normal to the membrane in the rz-plane. For anisotropic materials, the tangential displacements derivatives $u_{1\mathrm{n}}$ and $u_{2\mathrm{n}}$ are additionally added as dependent variables.

Stiffness in the Normal Direction

When membrane elements are used separately, not supported by other structural elements, a prestress is necessary in order to avoid a singularity. The unstressed membrane has no stiffness in the normal direction. It is the geometrically nonlinear effects (*stress stiffening*) which supply the out-of-plane stiffness. A prestress can be given either through initial stress and strain or through a tensile boundary load. Prestress is not necessary in cases where inertia effects are included in a dynamic analysis. A small prestress can, however, still be useful to stabilize the analysis in the initial state. In order to obtain the prestress effect, you must select **Include geometric nonlinearity** in the settings for the study step.

For a detailed discussion about dynamic analysis of prestressed structures, see Prestressed Structures.

When using membranes, also the prestress step must be geometrically nonlinear.

If you want to explicitly prescribe the stress field for a prestressed analysis rather than solving for it, you should not use the two study step procedure. In such a case, prescribe the stress field using an **Initial Stress and Strain**, **External Stress**, or **External Strain** node. Then add a separate **Eigenfrequency** or **Frequency Domain** study and select **Include Geometric Nonlinearity** in the settings for the study step.

Stabilization

When a membrane is unstable in its original configuration, you can use an automatic stabilization technique in order to avoid numerical problems until the external loads have created the necessary tension. To do that, you add a Stabilization node.

The effect is that a spring, acting in the direction perpendicular to the membrane is added. This spring is then removed using either an automatic scheme, or by a customized rule. When using the automatic version, you should increase the external load gradually, using at least three steps in order to ensure that no effect of the added spring remains.

Layered and Nonlayered Membranes

The Membrane interface includes several material models; Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, Hyperelastic Material, and External Stress–Strain Relation.

The fundamental difference between **Linear Elastic Material** and **Linear Elastic Material, Layered** is that in **Linear Elastic Material** the material properties are assumed to be constant through the thickness. In the **Linear Elastic Material, Layered** model, there is a numerical integration in the thickness direction, and it is also possible to store states, such as inelastic strains, at different through-thickness locations.

When the Composite Materials Module is available, the **Linear Elastic Material, Layered** model can be used to model multilayered membrane. This is the main use of this material model. It is, however, also used for the Thermal Expansion, Layered multiphysics coupling, even if there is just a single layer.

Since membranes are thin, the actual order of the layers in a multilayered membrane is not considered for the analysis.

For each layer, you have the option to set the resolution in the thickness direction. In a layered material, this is the **Mesh elements** property in the layer definitions. When working with a single layer material, then it is the **Mesh elements** property in the **Shell** property group. For membrane analysis, you can set this value to '1' since in-layer variations are not part of the theory.

Modeling with Beam Elements

In this section:

- About Beams
- In-Plane Beams
- 3D Beam
- Orientation of the Cross Section
- Geometric Variables
- Result Evaluation
- Modeling Stiffeners
- Nonrigid Joints

About Beams

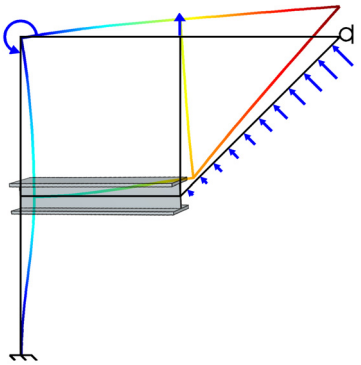
A *beam* is a slender structure that can be fully described by its cross-section properties such as area, moments of inertia, and torsional constant. Beams are the choice for modeling reinforcements in 3D solids and shell structures, as well as in 2D solids under the plane stress assumption. Naturally, they can also model latticeworks, both planar and three-dimensional.

Beams can sustain forces and moments in any direction, both distributed and on individual nodes. The beam's ends and interconnections can be free, simply supported, or clamped. In fact, the simplified boundary conditions are usually responsible for most of the difference that can be found between a beam solution and a full 3D solid simulation of the same structure. Point constraints on beams are well behaved, in contrast to the solid case, and it is possible to use discrete point masses and mass moments of inertia.

The Beam interface is based on the principle of virtual work. The resulting equation can equivalently be viewed as a weak formulation of an underlying PDE. The Beam interface uses special shape function classes to define stresses and strains in the beams using either Euler-Bernoulli or Timoshenko theory.

In-Plane Beams

Use the Beam interface in 2D to analyze planar lattice works of uniaxial beams.



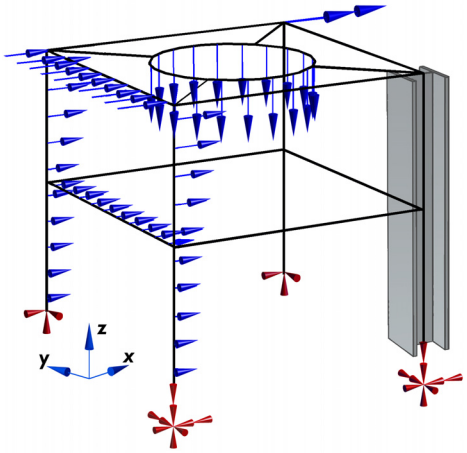
In-plane beams are defined on edges in 2D. They can be used separately or as stiffeners to 2D solid elements.

VARIABLES AND SPACE DIMENSIONS

The degrees of freedom (dependent variables) are the global displacements u and v in the global x and y directions and the rotation θ about the global z-axis.

3D Beam

Use the Beam interface in 3D to model three-dimensional frameworks of uniaxial beams.



3D beams are defined on edges in 3D. They can be used separately or as stiffeners to 2D solid or shell elements.

VARIABLES AND SPACE DIMENSIONS

The degrees of freedom (dependent variables) are the global displacements u, v, w in the global x, y, z directions and the global rotations θ_x, θ_y, and θ_z about the global x-, y-, and z-axes.

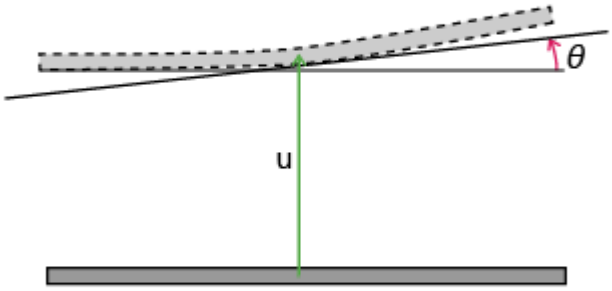


Figure 2-9: The degrees of freedom in the Beam interface.

Orientation of the Cross Section

For 3D beams, the orientation of the cross section in the plane perpendicular to the beam axis is an important property. The bending stiffness of most cross sections is highly anisotropic. Only a few cross sections, like circular pipes and square sections, have a stiffness that is the same in all directions. Even in this case, the section orientation will still usually be important for a correct evaluation and interpretation of stresses.

Any cross section has a local yz coordinate system, which correspond to the two principal area moments of inertia, I_{yy} and I_{zz}. I_{yy} is defined as the moment of inertia for bending around the y-axis,

$$I_{yy} = \int_A z^2 dA$$

Whether the local y-axis or z-axis is used as the stiffer direction is not important as such. For the built-in cross sections (H, T, C, hat) the convention is that $I_{zz} > I_{yy}$, so that the local y-axis is the stiffer direction.

There are two methods by which you can supply the cross-section orientation information. In either case, it is the orientation of the local y-axis that is described.

REFERENCE POINT

When you use the reference point input method, you give the coordinates of a point in space. The local y-axis is perpendicular to the beam axis and located in the plane described by the beam element and the given point.

ORIENTATION VECTOR

When using orientation vector input, you provide an approximate direction of the y-axis. This given vector is then adjusted in the plane given by the beam and vector, so that a y-axis perpendicular to the beam is obtained.

Once this direction has been established, it is possible to further rotate it around the beam axis. This is particularly useful for L-shaped sections, where the angle between a coordinate system aligned with the flanges and the principal axes often is provided in design tables.

See the documentation of the Section Orientation node for details about how the local directions are computed from the given input.

Geometric Variables

The beam formulations are based on the assumption that cross-section data are known. You can either enter them explicitly or, for a number of standard cross sections, get

them computed internally. In Table 2-5 contains a summary of the important geometric variables used in the Beam interface.

TABLE 2-5: GEOMETRIC VARIABLES IN BEAM INTERFACE.

PROPERTY	VARIABLE NAME	DESCRIPTION
A	`beam.area`	Cross-section area
I_{zz}	`beam.Izz`	Moment of inertia around local z-axis
e_z	`beam.ez`	Distance to shear center in local z direction
μ_y	`beam.muy`	Max shear stress factor in local y direction
κ_y	`beam.kappay`	Shear correction factor along local y-axis
r_{gy}	`beam.rgy`	Radius of gyration, local y direction
I_{yy}	`beam.Iyy`	Moment of inertia around local y-axis
e_y	`beam.ey`	Distance to shear center in local y direction
μ_z	`beam.muz`	Max shear stress factor in local z direction
κ_z	`beam.kappaz`	Shear correction factor along local z-axis
r_{gz}	`beam.rgz`	Radius of gyration, local z direction
J	`beam.J_beam`	Torsional constant
W_t	`beam.Wt`	Torsional section modulus
r_e	`beam.re`	Equivalent radius
	`beam.yBeam1` `beam.zBeam1`	Stress evaluation point 1, local coordinates
	`beam.yBeam2` `beam.zBeam2`	Stress evaluation point 2, local coordinates
	`beam.yBeam3` `beam.zBeam3`	Stress evaluation point 3, local coordinates
	`beam.yBeam4` `beam.zBeam4`	Stress evaluation point 4, local coordinates

Result Evaluation

The primary results in the Beam interface are the section forces: axial force, shear forces, bending moment, and twisting moment. The shear forces and bending moments are oriented along the principal directions as described in the Section Orientation node.

The formulation of the beam element is such that the variation of axial force, bending moments, and twisting moment is linear along the element, whereas the shear forces

are constant within each element. This means that the solution is exact with any discretization as long as there are no distributed transverse loads.

When looking at line plots or line graphs of section forces, it is important consider the type of averaging that is used between elements. In a frame structure, axial forces and shear forces are continuous as long as two adjacent beams have the same direction. If two beams, for example, meet a right angle, the shear force in one of them will be the axial force in the other. In this case, an averaging of section forces at the common point clearly would be wrong. Similarly, a twisting moment in one of the beams will be a bending moment in the next.

In the **Quality** section of the plot, you can control the level of averaging by changing the Smoothing method.

For details about the settings in the **Quality** section, see Entering Quality Settings for Plots and Plot Groups.

The default method is **Inside material domains**. In the Beam interface, two adjacent edges are considered as part of the same material domain only if they share both material and cross-section data. In many cases this gives an optimal level of smoothing since beams that meet at nonzero angles often have different cross sections.

You may however want to use **Inside geometry domains** instead as smoothing method. This means that there will be no smoothing at points where a connection may occur.

Since the stresses are functions of the section forces, the same reasoning applies to stress plots.

Evaluation of stresses in beam elements required special consideration since the stress field produced by various section forces have different distributions over the cross section. All that is known in the beam formulation is the peak value of each stress contribution. Details about how stresses are combined are given in the section Stress Evaluation. In general, the different contributions to the total stress are combined in a conservative manner. If you need to study the stress distribution over the cross section in detail, this can be done using the Beam Cross Section interface.

SHEAR FORCE AND MOMENT DIAGRAMS

A common representation of beam results is to draw diagrams of shear forces and bending moments on top of the structure. Such diagrams are generated as result templates, and you can add them from the **Result Templates** window. They are available as the group **Section Force Diagrams**.

The section force diagrams do not use the standard variables for shear forces and bending moments. These variables, like `beam.Tzl` and `beam.Myl`, are constant and linear, respectively, within each element, as an effect of the finite element formulation. Rather, a set of variables that are augmented with information about distributed loads is used. The names of these variables are `beam.Tzl_d` and `beam.Myl_d`, for example. In essence, a numerical integration of the loads is performed along each element up to the position where a value is requested. The augmented variables have a much smoother distribution than the standard section forces.

The effect is that you can display good section force diagrams also with a very coarse mesh. It should however be noted that the numerical integration of the loads has a limited resolution. This is not a problem for smooth distributed loads, but if you would add a discontinuous line load with an expression like `q0*(X>1.52[m])*(X<1.54[m])`, the load integration may fail to capture it accurately if the element length is an order of magnitude larger than the 0.02 m over which the load is distributed. This is seldom a problem in reality, since such a mesh can be considered as too coarse, and may not even capture the true resultant of such a load.

The section force diagrams are constructed using **Arrow Line** plots and you may need to tune them for your model. Here are some hints on how to work with the section force diagrams:

- The arrows are drawn in the direction of the action. In 3D, this means that `beam.Tzl_d` and `beam.Myl_d` are drawn in one plane, while `beam.Tyl_d` and `beam.Mzl_d` are drawn in the orthogonal plane.
- The coloring of the arrows is chosen so that red indicates a positive quantity and blue indicates a negative quantity.
- Maximum and minimum value markers are used in the plots. You may want to set a background color in the **Coloring and Style** section of the settings for the **Max/Min Line** node.
- The number of arrows is, as a default, set to 500. This may be to dense or too coarse for a certain structure.

Geometric Nonlinearity

For geometrically nonlinear studies with the Beam interface, a corotational formulation is used. It takes large displacements and rotations into account, but assumes small strains. This approach differs from what is used in the other structural mechanics interfaces.

The essence of the formulation is that each element in the mesh has a linear formulation within its own frame, but this frame can translate and rotate. The change in angle over a single element is assumed to be small. Thus, the number of elements to use in the mesh will depend on the expected curvature. This stands in strong contrast to linear beam analysis, where the only a single element is needed for each physical beam as long as there are no distributed loads.

A consequence of this formulation is that it is not possible to perform prestressed analysis with the Beam interface.

Modeling Stiffeners

One common use of beams is as stiffeners in shell structures. Ship hulls and aircraft fuselages, for example, are often built using this technique. An important property here is the offset between the shell midsurface and the centerline of the attached beam. To a large extent, it is the tension of the beam rather than its bending that provides the high stiffness. In order to model this effect, you can use two different approaches:

- Model the beam centerline and shell midsurface at their correct locations. Make sure that there are edges in the shell geometry placed as imprints under the stiffeners. The two sets of edges are then connected using suitable couplings.
- Draw only the "imprint" on the shell, and select these edges in the Beam interface. Connect the degrees of freedom in the two interfaces with a coupling that mathematically takes the offset into account.

In both cases, you use the Solid–Beam Connection multiphysics coupling, but with different settings. From the point of solution accuracy, the methods are equivalent for straight beams. Curved beams should always be modeled using the beam centerline in the geometry. The latter method is more convenient when applicable, but it has the drawback that the beams will be visualized as being located in the plane of the shell.

Both types of connections are shown in the example *Connecting Shells and Beams*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/shell_beam_connection**

Since the beam elements use special shape functions, a connection to another element type like a shell will not be conforming. Consider a beam attached to a shell element having the default discretization order (quadratic). There will then be two beam elements connected to each edge of a shell element. All displacements and rotations in the shell element are assumed to have a parabolic distribution. In the beam, however,

the axial displacements are linear, while the transverse displacements are represented by cubic polynomials. An effect of this is that there will be local aberrations in the forces transferred between the shell and the beam, even though the solution is correct in an average sense. Because of this, you may see local stress fluctuations.

Nonrigid Joints

As a default, all beams connected at a point are assumed to be rigidly coupled to each other. Sometimes, this is not the case in a true structure. One common case is the addition of hinges in a frame to avoid intrinsic stresses caused by for example thermal expansion or tolerance deviations.

To model such joints, you use the Beam End Release node. In that node, you can specify that some degrees of freedom are decoupled at a certain point.

USING EDGE GROUPS

If more than two beams are attached to such a point, the decoupling is however not unique. Consider a case where four beams meet at right angles, and that there is a hinge at the common point as shown in Figure 2-10. To model the hinge, the out-of-plane rotation is selected as a decoupled degree of freedom in the **Beam End Release** node.

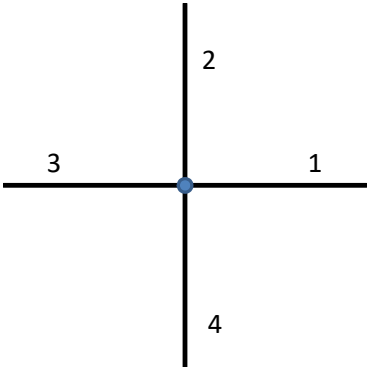


Figure 2-10: Original configuration.

There are several ways in which a hinge at the common connection point can be interpreted. Some examples are shown in Figure 2-11 to Figure 2-13. The default behavior is that all beams are disconnected in terms of rotation, as displayed in Figure 2-11.

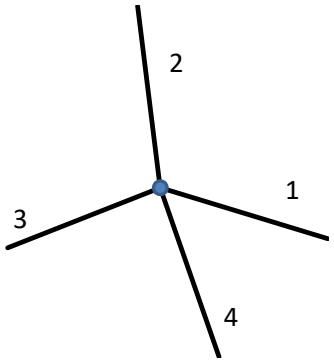


Figure 2-11: All beams are free to rotate relative to each other.

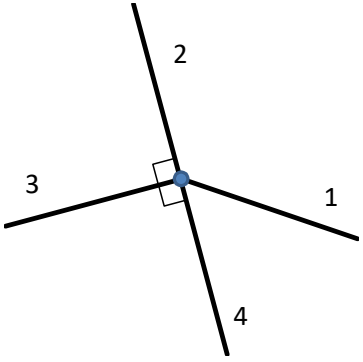


Figure 2-12: Beam 1 is free to rotate with respect to the other three beams.

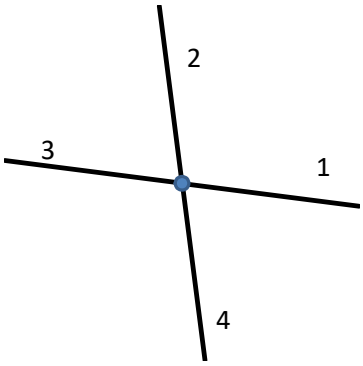


Figure 2-13: Beams 1 and 3 rotate as a common part with respect to beams 2 and 4.

If you want to describe nondefault types of connections, you need to add one or more Edge Group subnodes to the **Beam End Release** node. If just a single beam is disconnected, as in Figure 2-12, then you need a single edge group in which you select the other three beams (2, 3, and 4).

To describe the case in Figure 2-13, you add two edge groups. You then select beams 1 and 3 in the first one, and beams 2 and 4 in the second one.

Generally speaking, a beam assigned to an edge group will not have any degrees of freedom released with respect to other beams in the same group. A beam cannot belong to more than one edge group.

Stress Analysis of a Portal Crane: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/portal_crane**

FINITE ROTATIONS

In a geometrically nonlinear study, there may be finite rotations. In that case, the axis orientations along which degrees of freedoms are released will be updated based on the rotation of the beam itself. For the most common type of connections, this will be the intended behavior.

There are, however, some cases when the orientations can become ambiguous. Consider, for example, a 2D case where both the X-translation and the out-of-plane rotation are decoupled where two beams meet. This situation is shown in Figure 2-14. Since the rotation is now different in the two beams, so is the definition of a rotated X direction. Depending on the physical arrangement, the translational sliding motion

can follow either of the two connected beams. In the section **Edge Defining the Local Direction** in the settings for **Beam End Release**, you can select the edge that controls the rotation of the coordinate system under finite rotations.

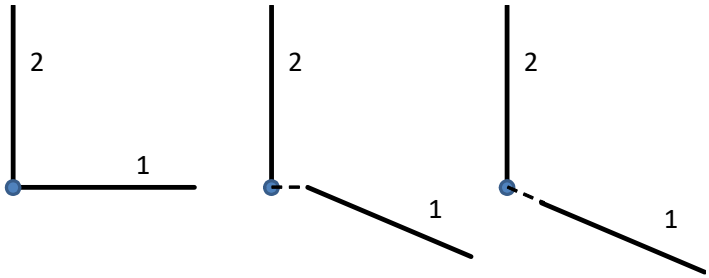


Figure 2-14: From left to right: Original configuration, deformed with X interpreted from beam 2, and deformed with X interpreted from beam 1.

Modeling with Truss Elements

Truss elements are elements that can only sustain axial forces. They have displacements as degrees of freedom. Truss elements are sometimes referred to as *bars* or *spars*. They live on boundaries in 2D and edges in 3D. Typical uses of truss elements are:

- Trusses
- Cables and wires, given that they are in tension only
- Reinforcement bars
- Two-point springs and dashpots. A special material model called Spring-Damper Material is available for this purpose.
- Thin 'strain gauges' attached to for example solid elements

Dependent Variables

The degrees of freedom (dependent variables) are the global displacements u, v, and w (3D only) in the global x, y, and z directions, respectively.

MODELING WIRES AND CABLES

You can use a Truss interface for modeling wires and cables, possibly sagging under gravity or other external loads. However, the Wire interface is specifically designed for this purpose, and should be the primary choice.

Below are some suggestions for how to model such structures efficiently using a Truss interface:

- It is generally recommended that you use first order shape functions when modeling wire-like structures. This is the default.
- If the cable is to be allowed to sag, you must not use the Straight Edge Constraint for those edges. When using first order shape functions, this constraint is disabled as default, so this is an issue only if you use higher order shape functions.
- Most cable problems are geometrically nonlinear. A wire which is not in tension is not numerically stable. Physically, it wrinkles in an unpredictable manner. In order to start the analysis, you either have to add an initial stress or some weak springs.
- If there are no line or volume loads, the wire is straight, as long as it is in tension. In this case, only one element is needed for the whole wire, since you can handle the zero-stiffness in compression in the material data. Enter the modulus of elasticity so

that it depends on the axial stress, and has very small value in compression. Such an expression could for example be `E0*(1-0.9999*(truss.en<0))`.

- In some problems, there are large deformations, but low tensile stresses. This would for example be the case if you model a wire hanging free under self-weight ('the catenary problem'). Such problems are numerically ill-conditioned, but can still be solved as long as you use linear shape functions for the truss elements, and use tight tolerances for the nonlinear solver.

Modeling Wires and Cables

Wires are thin one-dimensional structures with no bending stiffness. Ideal wires can only sustain tensile axial forces. Typically, wires act in a prestressed state. It is however also possible to model for example wires that are sagging under self-weight.

The wire elements have displacements as degrees of freedom and live on boundaries in 2D and edges in 3D. Typical uses of the wire elements are:

- Prestressed cables.
- Drive belts.
- Cables that hang free, subjected to for example gravity.

The Wire interface supports the same study types as the Solid Mechanics interface.

Dependent Variables

The degrees of freedom (dependent variables) are the global displacements u, v, and w (3D only) in the global x, y, and z directions, respectively.

MODELING WIRES AND CABLES

You use the Wire interface for modeling wires and cables, possibly sagging under gravity or other external loads. Below are some suggestions for how to model such structures efficiently:

- Most cable problems are geometrically nonlinear. The Wire interface will force any study to be geometrically nonlinear. A wire which is not in tension is not numerically stable. Physically, it wrinkles in an unpredictable manner. In order to start the analysis, you either have to add an initial stress or some weak springs.
- If there are no line or volume loads, the wire is straight. In this case, only one element is needed for the whole wire, since the force is constant.
- In some problems, there are large deformations, but low tensile stresses. This would for example be the case if you model a wire hanging free under self-weight ('the catenary problem'). Such problems are numerically ill-conditioned. They can be solved, but you may have to use tight tolerances for the nonlinear solver and allow many iterations.

Modeling with the Phase Field in Solids Interface

In this section:

- Introduction
- Dependent Variables
- Phase–Field Damage Modeling

Introduction

Phase field modeling is used in various areas of physics for solving problems involving moving interfaces. In solids, moving interfaces of interest can be cracks, damage, or grain boundaries. Instead of tracking the evolution of sharp interface in a domain (for example, the propagation of a crack), phase field models use regularization techniques to represent sharp interfaces with a diffuse order parameter that varies smoothly over the whole domain. The phase field regularization introduces an *internal length scale*, l_{int}, that recovers the sharp interface in the limit $l_{\text{int}} \to 0$. Therefore, the internal length scale in a phase field model essentially imposes a constraint on the maximum mesh size that can be used in the vicinity of a moving interface.

Dependent Variables

The dependent variable is the dimensionless phase field ϕ with the default variable name `phi`. By default, a single phase field variable is included when the interface is added to a component, but multiple phase field variables can be added. When modeling damage and fracture in solids, the phase field is a bounded quantity that represents a smooth transition from undamaged ($\phi = 0$) to fully damaged material ($\phi = 1$).

The governing equations of the phase field are formulated on the material frame in a total Lagrangian formulation consistent with the Solid Mechanics interface, see the sections Material and Spatial Coordinates and Lagrangian Formulation in the *Structural Mechanics Theory* chapter.

Phase–Field Damage Modeling

An important use case of phase-field modeling in solids is the prediction of damage initiation and crack propagation in structural components. Here, the driving force for the phase field evolution depends on, for example, the state of stress in the solid, and, vice versa, the evolution of the phase field determines the degradation of the solid properties. Such bidirectionally coupled problems can be solved with the Phase-Field Damage (Multiphysics Coupling) node.

Modeling with the Transport in Solids Interface

In this section:

- Introduction
- Dependent Variables
- Material and Spatial Variables
- Mass Transfer Variables

Introduction

Transport processes in solids are mainly driven by diffusion, and, unlike in fluids, convection is absent. Fick's law of diffusion is a widely employed model for characterizing mass transport in solids. It posits that diffusion is primarily driven by the concentration gradient of the species. In the Transport in Solids interface, mass transport can be specified by Fick's law or by the gradient of the chemical potential.

Dependent Variables

The dependent variables are referred as to **Transported Quantities**, and their SI unit can be customized. The default mass concentration per unit of (undeformed) solid volume can be expressed in the SI unit of kilograms or moles per cubic meter. To model the transport of ions, or holes and electrons in semiconductors, the *number density* is also available.

By default, one species is available in the Transport in Solids interface, but multiple species can be added. The dependent variable can be retrieved by using the physics interface name (the prefix). As an example, you can plot the concentration of species `c` with the default physics interface name `ts`, so this variable reads `ts.c`.

The difference between material and spatial frame variables, like concentration and density, stems from the volume that is being referred to (for instance kilogram per cubic meter). A material frame variable uses the undeformed (reference) volume whereas a spatial frame variable uses the deformed (current) volume of the solid where it is transported. These are the so-called material and spatial frames, respectively. They are related through the determinant J of the deformation gradient as follows

$$c_{\mathrm{mat}} = Jc_{\mathrm{spa}}$$

here, c_{mat} refers to the mass per reference unit volume, and c_{spa} as the mass per deformed unit volume.

For more information see the Material and Spatial Coordinates and Lagrangian Formulation sections in the *Structural Mechanics Theory* chapter.

Material and Spatial Variables

DIFFUSION

The diffusion coefficient in solids can be given in the reference configuration (undeformed configuration, or material frame) or in the deformed configuration (current configuration, or spatial frame). The relation between these reads

$$D = JF^{-1}dF^{-T}$$

here, D is the diffusion coefficient in the reference configuration, and d represents the diffusion coefficient in the deformed configuration. The variables F and J are the deformation gradient and its determinant.

A constant diffusion coefficient D given in material frame implies that its corresponding value in spatial frame, d, depends on the deformation

$$d = J^{-1}FDF^{T}$$

The frame in which the diffusion coefficient is given can be specified in the **Advanced** section. From the **Frame** list select **Reference configuration** or **Deformed configuration** to specify a material or spatial diffusion.

SOURCES

The source term is a scalar quantity that is given per unit volume, so the same considerations apply as with the dependent variables, see Dependent Variables. The frame in which the source is given can be selected in the **Advanced** section.

DOMAIN FLUXES

Fluxes are vectors that can be expressed in the material frame or the spatial frame. These are related as follows,

$$\Gamma = JF^{-1}\gamma$$

here, Γ is the flux in the undeformed configuration, γ is the flux in the deformed configuration, F is the deformation gradient, and J is its determinant.

The **External Flux** node in the Transport in Solids interface allows adding additional domain fluxes. The frame in which the domain flux is given can be selected in the **Advanced** section.

NORMAL BOUNDARY FLUXES

When specifying normal boundary fluxes as boundary conditions, it is assumed that the normal flux follows the deformed boundary. The normal flux is related to the material frame and spatial frame fluxes as

$$-\Gamma \cdot \mathbf{N} = \Gamma_n = J_A \gamma_n = -J_A \gamma \cdot \mathbf{n}$$

Here, $\mathbf{N}$ is the normal to the undeformed boundary, $\mathbf{n}$ is the normal to the deformed boundary, and the variable J_A is the ratio between the deformed area and the reference area. By default, the normal flux γ_n is set to the deformed configuration (spatial frame), but it is also possible to specify the normal flux in the reference configuration (material frame), Γ_n. The frame in which the normal flux is given can be specify in the **Advanced** section.

Mass Transfer Variables

This section lists the predefined variables to evaluate fluxes used in the mass balance. All the variables names begin with the physics interface name (the prefix) and end with the species that you want to inspect. By default, the Transport in Solids interface prefix is `ts` and the default dependent variable name is `c`. As an example, you can access the variable named `dflux` for the dependent variable `c` using `ts.dflux_c`

TABLE 2-6: MASS TRANSFER VARIABLES.

VARIABLE	NAME	GEOMETRY ENTITY LEVEL
dflux	Diffusive Flux	Domain
dfluxMag	Diffusive Flux Magnitude	Domain
domflux	Domain Flux	Domain
extflux	External Flux	Domain
extfluxMag	External Flux Magnitude	Domain
tflux	Total Flux	Domain

TABLE 2-6: MASS TRANSFER VARIABLES.

VARIABLE	NAME	GEOMETRY ENTITY LEVEL
tfluxMag	Total Flux Magnitude	Domain
bndFlux	Boundary Flux	Boundary
ndflux	Normal Diffusive Flux	Boundary
ntflux	Normal Total Flux	Boundary

DOMAIN FLUXES

In addition to the flux variables, there are also predefined flux magnitudes (scalar quantities) in the domains.

Diffusive Flux

The diffusive flux variable, `dflux`, is defined by Fick's law:

$$\text{dflux} = -D\nabla c$$

Diffusive Flux Magnitude

The diffusive flux magnitude, `dfluxMag`, is the magnitude of the diffusive flux in material frame, defined as the square root of the sum of the diffusive components squared:

$$\text{dfluxMag} = \sqrt{\text{dflux}_X^2 + \text{dflux}_Y^2 + \text{dflux}_Z^2}$$

Domain Flux

The variable `domflux` is used to calculate fluxes when the **Compute boundary fluxes** checkbox is cleared. The flux variables `ndflux` and `ntflux` are then computed from the dependent variables using extrapolation. The domain flux variable `domflux` is the defined as the diffusive flux times the out-of-plane geometry extension (out-of-plane thickness for 2D, or cross-sectional area for 1D or 1D axisymmetric):

$$\text{domflux} = \text{dflux} \cdot d_z$$

For more information about the boundary fluxes settings, see Computing Accurate Fluxes in the *COMSOL Multiphysics Reference Manual*.

External Flux

The external flux is a user input in the **External Flux** node.

External Flux Magnitude

The external flux magnitude is the magnitude of the external flux:

$$\text{extfluxMag} = \sqrt{\text{extflux}_X^2 + \text{extflux}_Y^2 + \text{extflux}_Z^2}$$

Total Flux

The total flux is the sum of the diffusive flux and external fluxes:

$$\text{tflux} = \text{dflux} + \text{extflux}$$

Total Flux Magnitude

The total flux magnitude is the magnitude of the total flux:

$$\text{tfluxMag} = \sqrt{\text{tflux}_X^2 + \text{tflux}_Y^2 + \text{tflux}_Z^2}$$

BOUNDARY FLUXES

All domain fluxes (vector quantities) are also available as boundary fluxes. The boundary fluxes are equal to the value of the flux in the adjacent domain. Additionally, normal boundary fluxes are also available on boundaries.

Boundary Flux

On exterior boundaries, `bndFlux` is defined as

$$\text{bndFlux} = -\text{dflux}(c)/d_z$$
$$\text{bndFlux} = -\text{uflux}(c)/d_z$$

whether the adjacent domain is on the downside or on the upside. The up and downside is defined by the surface normal, which points from downside to the upside. On interior boundaries, `bndFlux` is the mean of the fluxes on both sides:

$$\text{bndFlux} = 0{,}5(\text{uflux}(c) - \text{dflux}(c))/d_z$$

The variable `bndFlux` is defined using the Boundary Flux Operators: uflux and dflux. These operators are used when the **Compute boundary fluxes** checkbox is selected, see Computing Accurate Fluxes in the *COMSOL Multiphysics Reference Manual*.

Normal Diffusive Flux

The variable `ndflux` is defined as

$$\text{ndflux} = \text{dflux} \cdot \mathbf{n}$$

Normal Total Flux

The variable `ntflux` is defined as

$$\text{ntflux} = \text{bndFlux}$$

Coupling Different Element Types

In this section:

- Introduction to the Element Types
- Coupling Techniques

Introduction to the Element Types

When modeling some structures, the optimal idealization is a mixture of different element types. Some examples are:

- Structures that are thin in large regions, but more three-dimensional at certain locations. A mixture of solids and shells can then significantly reduce the model size.
- Plates or shells having beams as stiffeners.
- Truss elements acting as reinforcement bars in a concrete structure.
- A thin layer of one material on top of another material. In this case, an idealization with shells or membranes covering the boundary of a solid can be useful.

When several physics interfaces are added in COMSOL Multiphysics, the default is always that each physics interface has individual and unique degrees of freedom. In structural mechanics, the first physics interface has the displacement variables (`u`, `v`, `w`), then the second physics interface has (`u2`, `v2`, `w2`), and so on. This means that the physics interfaces initially are independent, even when defined on the same geometrical part. To get the intended interaction requires that a coupling is established between the physics interfaces.

Various methods to couple different element types are discussed in this section.

Coupling Techniques

The following basic techniques to connect physics interfaces with displacement degrees of freedom are discussed in this section:

- Renaming Degrees of Freedom
- Using Customized Coupling Features
- Using General Coupling Operators

RENAMING DEGREES OF FREEDOM

The easiest coupling method is to rename the displacement degrees of freedom so that these are the same for all interacting structural mechanics interfaces. This is sufficient, for example, when using membranes as cladding on a solid boundary or when using truss elements as reinforcement bars in a solid.

In the Beam, Pipe Mechanics, Shell, and Plate interfaces, the deformation is described also by a set of rotational degrees of freedom. In the general case, these degrees of freedom interact with the translational degrees of freedom in a connection.

In some special cases — for example, when a thin shell acts as cladding on a solid — it is sufficient to make the degree of freedom names for the displacements common; the rotational degrees of freedom are not important. If, however, a shell edge is connected to a solid, it acts as a "hinge", which in most cases is not the intended behavior. You then need to use the more sophisticated techniques described in the following.

- The default shape functions in the Solid Mechanics interface are of the serendipity type, whereas Lagrange shape functions are used in the Shell interface. If you are placing a shell element on the boundary of a solid element, you must select Lagrange shape functions also in the Solid Mechanics interface so that the two physics interfaces share the same node points.
- The shape functions used in the Beam and Pipe Mechanics interfaces have special properties, and a beam cannot have the same degrees of freedom as another physics interface if the same edge is shared.
- The representation of rotations differs between the Shell and Plate interfaces (displacement of normal) and the Beam and Pipe Mechanics interfaces (rotation angle). It is therefore not possible to use common degree of freedom names to couple the rotational degrees of freedom.

USING CUSTOMIZED COUPLING FEATURES

There are a number of built-in couplings, by which you can add connections that are nontrivial to set up manually:

- Shell Edge to Solid Boundary (3D, 2D axisymmetry)
- Shell Boundary to Solid Boundary (3D, 2D Axisymmetry)
- Membrane Boundary to Solid Boundary (3D, 2D Axisymmetry)
- Beam Point to Solid Boundary (2D)

- Beam Point to Solid (3D)
- Beam Edge to Solid Boundary (2D)
- Beam Edge to Solid Boundary (3D)
- Beam Edge to Shell Edge (3D)
- Beam Point to Shell Boundary (3D)
- Beam Point to Shell Edge (3D)
- Pipe Point to Solid Boundary (3D)
- Pipe Point to Shell Edge (3D)
- Embedded Reinforcement

Shell Edge to Solid Boundary (3D, 2D axisymmetry)
A shell edge can be coupled to the face of a solid by adding a Solid–Thin Structure Connection multiphysics coupling. In the settings, set **Connection type** to **Solid boundaries to shell edges**. This situation typically occurs when you want to make a transition from a thin region to one that is thicker. Usually, shell assumptions should be valid on both sides of the transition. The solid geometry is expected to have the same thickness as the thickness given in the Shell interface.

You can choose between two different formulations, by setting **Method** to either **Rigid** or **Flexible**. The flexible version is significantly more accurate locally at the connected solid boundary, but it comes with a cost in terms of some extra degrees of freedom. Also, this method requires a large enough number of degrees of freedom in the thickness direction of the solid. For second-order elements, typically three elements are required.

Shell Boundary to Solid Boundary (3D, 2D Axisymmetry)
A shell can also be coupled to a solid by adding a Solid–Thin Structure Connection multiphysics coupling with **Connection type** set to **Shared boundaries** or **Parallel boundaries**. This connection is used to add a shell on top of a solid as a 'cladding'. It is possible to include an offset distance. The boundaries may be coincident or parallel.

Membrane Boundary to Solid Boundary (3D, 2D Axisymmetry)
A membrane can be coupled to a solid by adding a Solid–Thin Structure Connection multiphysics coupling with **Connection type** set to **Shared boundaries**. When the thin structure is a membrane, this is the only available connection type. It is used to add a membrane on top of a solid as a 'cladding'.

Beam Point to Solid Boundary (2D)

A beam in 2D can be coupled to a solid by adding a Solid–Beam Connection multiphysics coupling. In the settings, set **Connection type** to **Solid boundaries to beam points**. This connection is intended for modeling a transition from a beam to a solid, where beam assumptions are valid on both sides of the connection.

You can choose between two different formulations, by setting **Method** to either **Rigid** or **Flexible**. The flexible version is significantly more accurate locally at the connected solid boundary, but it comes with a cost in terms of some extra degrees of freedom. Also, this method requires a large enough number of degrees of freedom in the thickness direction of the solid. For second-order elements, typically three elements are required.

Beam Point to Solid (3D)

A beam in 3D can be coupled to a solid by adding a Solid–Beam Connection multiphysics coupling. In the settings, set **Connection type** to either **Solid boundaries to beam points, general**. or **Solid boundaries to beam points, transition**. These two couplings are fundamentally different.

The **Solid boundaries to beam points, general** connection is used for modeling a beam with one end "welded" to the face of the solid. You can specify the size of the area on the solid boundary that is connected to the endpoint of the beam in several different ways.

The **Solid boundaries to beam points, transition** coupling is intended for modeling a transition from a beam to a solid where beam assumptions are valid on both sides of the connection. Thus, the geometry of the solid at the transition should match the cross-section data given to the beam.

This connection type can include warping of the solid cross section. In order to compute the warping properties, an extra PDE is solved over the cross-section boundaries. To improve the performance, you should preferably solve for these variables once in a separate stationary study or study step. In that study step, clear all physics interfaces except the **Solid–Beam Connection** multiphysics coupling in the **Physics and Variables Selection** section.

You can manually control whether to include warping or not. If not included, the setup of the solver sequence is simplified, but there will be significant stress disturbances close to the connection boundaries if the cross section is susceptible to warping.

There are four warping variables: one named `Warping function` and three named `Warping constant`. In the successive study steps, you need to manually suppress

them. You can do so under the **Dependent Variables** node, where you first set **Defined by study** step to **User Defined**. Then, for each of these four variables, clear the **Solve for this field** checkbox.

Beam Edge to Solid Boundary (2D)

A beam in 2D can also be coupled to a solid by adding a Solid–Beam Connection multiphysics coupling with **Connection type** set to **Shared boundaries** or **Parallel boundaries**. This connection is used for adding a beam on top of a solid as a "cladding". It is possible to include an offset distance. The boundaries may be coincident or parallel.

Beam Edge to Solid Boundary (3D)

A beam in 3D can also be coupled to a solid by adding a Solid–Beam Connection multiphysics coupling with **Connection type** set to **Solid boundaries to beam edges**. This connection is used for adding a beam that is "welded" along the surface of the solid.

Beam Edge to Shell Edge (3D)

A beam can be coupled to a shell by adding a Shell–Beam Connection multiphysics coupling with **Connection type** set to either **Shared boundaries** or **Parallel boundaries**. This connection is used for adding beams as stiffeners to shells. The edges may be coincident or parallel. It is possible to prescribe that the beam has an offset from the shell when a coincident edge is used.

Beam Point to Shell Boundary (3D)

A beam can be coupled to a shell by adding a Shell–Beam Connection multiphysics coupling with **Connection type** set to **Shell boundaries to beam points**. This connection is used for modeling a beam with one end "welded" to the face of the shell. You can specify the size of the area on the shell boundary that is connected to the end point of the beam in several different ways.

Beam Point to Shell Edge (3D)

A beam can be coupled to a shell by adding a Shell–Beam Connection multiphysics coupling with **Connection type** set to **Shell edges to beam points**. This connection is used for modeling a beam with one end "welded" to the edge of the shell. You can specify the part of the shell edge that is connected to the end point of the beam in several different ways.

Pipe Point to Solid Boundary (3D)

A pipe in 3D can be coupled to a solid by adding a Structure–Pipe Connection multiphysics coupling. The coupling is intended for modeling a transition from a pipe to a solid where beam assumptions are valid on both sides of the connection. Thus, the

geometry of the solid at the transition should match the cross-section data given to the pipe. The connection assumes that the pipe cross section is circular; if another cross section is used, it is converted to an equivalent circular cross section. This means that warping is not considered.

The connection can be considered an extension of the **Solid boundaries to beam points, transition** coupling in Solid–Beam Connection to also account for radial deformation of the pipe caused by the fluid pressure and the temperature difference over the cross section.

Pipe Point to Shell Edge (3D)

A pipe in 3D can be coupled to a shell by adding a Structure–Pipe Connection multiphysics coupling. The coupling is intended for modeling a transition from a pipe to a shell where beam assumptions are valid on both sides of the connection. Thus, the geometry of the shell at the transition should match the cross-section data given to the pipe. The connection assumes that the pipe cross section is circular; if another cross section is used, it is converted to an equivalent circular cross section. This means that warping is not considered.

The connection can be considered an extension of the **Shell edges to beam points** coupling in Shell–Beam Connection to also account for radial deformation of the pipe caused by the fluid pressure and the temperature difference over the cross section.

Embedded Reinforcement

Lower dimension structural elements can be connected to a solid domain by adding an Embedded Reinforcement multiphysics coupling. This connection supports coupling truss, wire, beam, and membrane elements to solid elements. The connection can either be rigid, or made by attaching springs between points on the embedded structure and points in the solid. A more detailed discussion about this type of modeling is given in Modeling Embedded Structures and Reinforcements.

The underlying theory and more details about the built-in couplings can be found in

- Connection Between Shells and Solids
- Connection Between Shells and Beams
- Embedded Elements

- Examples of all types of couplings between shells and beams are shown in *Connecting Shells and Beams*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/shell_beam_connection**
- An example of couplings between shells and solids is shown in *Connecting Shells and Solids*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/shell_solid_connection**
- An example of couplings between beams and solids is shown in *Connecting Beams and Solids*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/beam_solid_connection**

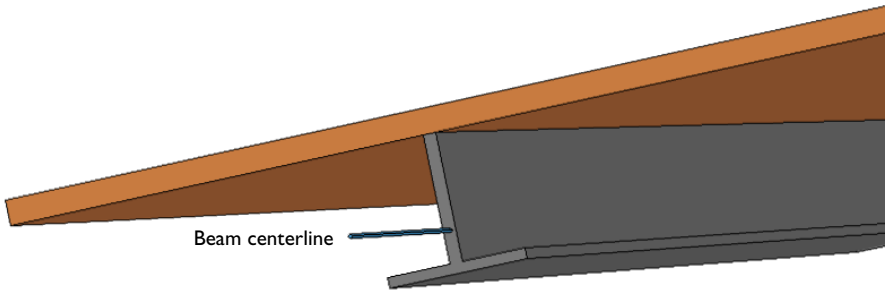
USING GENERAL COUPLING OPERATORS

The most general method of connecting parts modeled with different physics interfaces is by using a **General Extrusion** operator. In this case the parts need not even be in contact, so the connection is an abstraction.

An example could be a shell stiffened by beams. In practice, you would probably use the built-in coupling described in Beam Edge to Shell Edge (3D) for this case, but the example shows the principles.

In structure like this, the beam is usually placed at one side of the shell, so that the centerline of the beam and the midsurface of the shell do not coincide. This difference

must be taken into account, so the edges representing the beam are geometrically disconnected from the midsurface of the shell.



Mathematically, the connection between the beam and the shell can be expressed as

$$\begin{cases} \mathbf{u}_{\text{beam}} = \mathbf{u}_{\text{shell}} + \phi \times (\mathbf{X}_{\text{beam}} - \mathbf{X}_{\text{shell}}) \\ \phi_{\text{beam}} = \phi_{\text{shell}} \end{cases}$$

or equivalently as

$$\begin{cases} \mathbf{u}_{\text{beam}} = \mathbf{u}_{\text{shell}} + ((\mathbf{X}_{\text{beam}} - \mathbf{X}_{\text{shell}}) \cdot \mathbf{n})\mathbf{a} \\ \phi_{\text{beam}} = \phi_{\text{shell}} \end{cases}$$

Here, ϕ is the rotation vector, which contains the rotational degrees of freedom in the Beam interface. The rotation vector is also available as a variable in the Shell interface, where it is derived from the rotational degrees of freedom $\mathbf{a}$. The shell normal is denoted by $\mathbf{n}$. Small rotations are assumed.

To create the coupling:

1 Add a **General Extrusion** node under **Definitions** > **Nonlocal Couplings**. Select the line on the shell midsurface as source. Enter data in the **Destination Map**.

2 Add a Prescribed Displacement/Rotation node in the Beam interface and select the corresponding edge.

3 Enter data for the prescribed displacements and rotations, for example `genext1(u)+genext1(shell.thy)*zdist`, where `zdist` is some expression defining the distance from the beam axis to the shell midsurface.

Because a shell does not have a valid rotation degree of freedom around its normal, the rotation of the beam should not be connected in that direction.

In the *COMSOL Multiphysics Reference Manual*:

- Nonlocal Couplings and Coupling Operators and General Extrusion
- About Nonlocal Couplings

Modeling Embedded Structures and Reinforcements

Lower dimension structural elements such as trusses, beams, and membranes can be embedded into a solid domain by adding an Embedded Reinforcement multiphysics coupling. This modeling technique is intended for efficient modeling of thin structures in solids, where it is unfeasible or not necessary to resolve the geometry of the thin structure. Some typical use cases include:

- Reinforcement bars in concrete structures
- Piles or sheets in soil
- Fibers in composite materials
- Embedded sensors in a structure. Embedding sensors can either be used to model a real sensor, or as a modeling technique to extract, for example strain in some predefined location and direction in a solid domain.

The connection between the embedded structure and the solid can be set up using different techniques; either a rigid connection by adding constraints between the points in the respective interface, or by adding springs. When a spring type connection is used, the connection can be linear or nonlinear. Using a nonlinear connection can be important when modeling reinforcement in structures, especially when predicting failure and postfailure behavior.

The multiphysics coupling is intended for situations where the volume of the embedded structure is small compared to the solid domain. Hence, no compensation is made for the mass of the small structure. If a detailed model of the connection and composite structure is required, the geometry should include the interface between the embedded structure and the solid. The connection should then be modeled using

some other coupling technique as discussed in Coupling Different Element Types, or using Contact Modeling.

When the **Embedded structure** is a Beam interface and a **Rigid** connection type is used, the displacements are only constrained to the solid domain at the mesh vertices of the beam elements. This means that higher-order variations of the displacement field between mesh vertices is allowed. The **Spring** connection will, however, constrain also the higher-order variations of the displacement field.

The functionality provided by the **Embedded Reinforcement** multiphysics coupling to some extent overlaps with other couplings available to different structural mechanics interfaces. For example, both the **Embedded Reinforcement** and the Solid–Thin Structure Connection can be used to attach a membrane as a cladding on a solid domain. An important difference is, however, that the **Solid–Thin Structure Connection** will add constraints on the solid domain, while the **Embedded Reinforcement** will add constraints on the membrane. Depending on the mesh, the results may differ significantly.

The same difference applies when comparing the **Embedded Reinforcement** to a Solid–Beam Connection on shared edges.

GEOMETRY AND MESHING

When modeling embedded structures, it is sometimes easier to form the geometries of the coupled interfaces from an assembly. In the finalization step of the geometry sequence, you should select **Form an assembly** in the **Action** list. This will also put less restrictions on the mesh of the respective geometry. If **Form Union** is selected in the **Action** list, the same mesh will be shared by both interfaces; which has to respect and resolve the geometry of the embedded structure.

While there is no strict restriction on the mesh size when modeling embedded structures in an assembly; a good practice is that the mesh used for the two coupled interfaces is of approximately the same size, or that the mesh of the embedded structure is slightly finer. Use the same shape order in both interfaces when modeling embedded structures that share the geometry and mesh with the solid domain, that is, when **Form Union** is selected in the **Action** list.

In some situations, the connection can be sensitive to mesh elements of the embedded structure that are partially within the solid domain. This is especially sensitive for the penalty connection, where the weak equations are evaluated at the integration points of the embedded mesh element. A slight shift in the mesh can cause an integration point to either be inside or outside the solid domain.

PENALTY FACTORS

When using the spring-type connection, the accuracy of the connection is controlled by the spring stiffness matrix. The components of this matrix are most naturally represented in the local coordinate system of the embedded structure, for which only the diagonal components are of interest. Mathematically, the spring stiffness can be viewed as penalty factors that must be sufficiently large to accurately enforce the regularized constraint but not so large that the overall stiffness matrix becomes ill conditioned. This means that, in most cases, the spring stiffness in only a numerical parameter. However, there can be situations where the interface between the embedded structure and the solid domain has a measurable stiffness, for example, if the connection is used to idealize an interface with a finite, but small, thickness such as a glue layer. In such cases, the spring stiffness will have a real physical interpretation.

The default expression for the spring stiffness of the connection is derived from the stiffness and cross-sectional properties of the mesh element of the embedded structure. The available spring connection types and the corresponding default expressions for the spring stiffness are summarized in Table 2-7. All spring stiffness components have the same default expressions.

TABLE 2-7: DEFAULT EXPRESSION FOR THE SPRING STIFFNESS.

CONNECTION TYPE	SPRING CONSTANT PER REFERENCE LENGTH	SPRING CONSTANT PER REFERENCE SURFACE AREA
Truss	`1e3*truss.Eequ* truss.area/h^2`	`1e3*truss.Eequ* truss.perimeter/h^2`
Truss, Explicit Dynamics	`1*truss.Eequ* truss.area/h^2`	`1*truss.Eequ* truss.perimeter/h^2`
Wire	`1e3*wire.Eequ*beam.area/h^2`	`1e3*wire.Eequ* wire.perimeter/h^2`
Beam	`1e3*beam.Eequ*beam.area/h^2`	N/A
Membrane	N/A	`1e5*mbrn.Eequ* mbrn.d/h^2`

LOCAL ORIENTATIONS

Many quantities used by the connection, such as the spring stiffness and constitutive models, are most naturally represented in the local coordinate system of the embedded structure. For beam and membranes, the multiphysics coupling picks up the local coordinate system defined by the physics interface. However, in the Truss or Wire interfaces, only the direction of the local edge tangent $\mathbf{t}_{\mathrm{le}}$ is defined. For the multiphysics coupling, two transverse direction are also needed when using a spring connection. Thus, the multiphysics coupling defines a local coordinate system by assuming that $\mathbf{t}_{\mathrm{le}}$ is the normal to a plane. The actual directions of the two in-plane tangents are not important, and it is assumed that the first tangent points in the global z direction. Hence, the first transverse direction is

$$\mathbf{t}_1 = \frac{\mathbf{e}_z - (\mathbf{t}_{\mathrm{le}} \cdot \mathbf{e}_z)\mathbf{t}_{\mathrm{le}}}{\left\| \mathbf{e}_z - (\mathbf{t}_{\mathrm{le}} \cdot \mathbf{e}_z)\mathbf{t}_{\mathrm{le}} \right\|}$$

and the second transverse direction is

$$\mathbf{t}_2 = \mathbf{t}_{\mathrm{le}} \times \mathbf{t}_1 = \frac{\mathbf{t}_{\mathrm{le}} \times \mathbf{e}_z}{\left\| \mathbf{e}_z - (\mathbf{t}_{\mathrm{le}} \cdot \mathbf{e}_z)\mathbf{t}_{\mathrm{le}} \right\|}$$

where $\mathbf{e}_z$ is the base vector that points in the global z direction. Since no distinction is made between $\mathbf{t}_1$ and $\mathbf{t}_2$, only a single transverse spring constant, k_{t}, has to be entered when defining the spring stiffness matrix.

BOND SLIP MODELS

It is possible to include a bond slip model of the interface when modeling the embedded structure connection with a nonlinear spring. This nonlinear behavior follows a plasticity model, where the relative displacement between the two coupled interfaces is additively decomposed into an elastic displacement and a plastic displacement, or slip. Modeling bond slip is limited to small sliding only, since the mapping between the embedded structure and the solid domain is made in the reference configuration.

The slip is defined using a local constitutive model. This local model adds a set of internal degrees-of-freedom that are solved for and stored in the model at each Gauss point of the embedded structure. Typically, these internal variables include the relevant components of the slip vector $\mathbf{u}_{\mathrm{l,p}}$, the accumulated slip u_{pe}, and the friction dissipation density W_{p}. The last variable is only added if the **Calculate dissipated energy** checkbox is selected in the **Energy Dissipation** section. The internal variables used by the

Embedded Reinforcement multiphysics coupling are shown as separate fields under **Dependent Variables** node in the solver sequence.

The resistance to slide in the bond slip model is determined by the cohesion c, which can be a function of any variable or field present in the model; by adding a generic expression to the initial cohesion c_0. A built-in hardening model with respect to u_{pe} can be also be added, in which case $c = c_0 + c_{\mathrm{h}}$, where c_{h} describes some hardening function.

In many applications the bond slip behavior of the interface is described by a Coulomb type friction. However, the bond slip model by default has no dependence on the normal force acting on the interface since the connection has no formal direction of the normal of the interface. A Coulomb type friction model can still be considered if a good estimate of the normal force can be found by adding such a dependence to c_0. If, for example, the pressure in the solid domain can be considered as the normal pressure acting on the interface, use an expression like `0.1*`*`<tag>`*`.ExtCplOp(solid.p)` to define c_0 with a friction coefficient equal to 0.1. Here *`<tag>`* is the tag of the multiphysics coupling and *`<tag>`*`.ExtCplOp(`*`expr`*`)` is the operator that maps the expression from the solid domain to the embedded structure.

SOLVER SETTINGS

When adjusting the settings of the solver sequence for a model that includes an **Embedded Reinforcement** multiphysics coupling, make sure that the dependent variables of the coupled interfaces are solved in the same group. Also, note that the fields related to the bond slip model should be included in the same solver group as well. This is only a potential issue if a segregated solver is used, and it is handled automatically by the default solver suggestion.

Applying Loads

An important aspect of structural analysis is the formulation of the forces applied to the modeled structure. You can use custom expressions, predefined or user-defined coordinate systems, and even variables from other modeling physics interfaces.

Loads can be applied in the structural mechanics interfaces on the body, face, edge, or point levels. You can also apply loads to special features like **Rigid Material** or **Rigid Connector**. There is also an option to apply point loads to given coordinates, which do not have to coincide with a geometrical point or a node point.

In this section:

- Units, Orientation, and Visualization
- Load Cases
- Singular Loads
- Moments in the Solid Mechanics Interface
- Pressure
- Acceleration Loads
- Temperature Loads — Thermal Expansion
- Hygroscopic Swelling
- Total Loads
- Loads with a Spatial Variation
- Variables for Loads

Units, Orientation, and Visualization

USING UNITS

Enter loads in any unit, independently of the base SI unit system in the model, because COMSOL automatically converts any unit to the base SI unit system. To use the feature for automatic unit conversion, enter the unit in square brackets, for example, `100[lbf/in^2]`.

The exception is random vibration analysis. In that case, no automatic unit conversions are available, so you must enter loads in the base units of the model.

PREDEFINED AND CUSTOM COORDINATE SYSTEMS

Different predefined coordinate systems are available when materials or boundary conditions are specified. There is always the global coordinate system. Depending on the dimensionality of the part being worked with, there can also be predefined

coordinate systems such as the local tangent and normal coordinate system for boundaries.

Custom coordinate systems are also available and are useful, for example, to specify a load in any direction without splitting it into components. From the **Definitions** toolbar, select a **Coordinate System** ( ) from the menu.

Some coordinate systems can have solution dependent axis directions. If you use such a system for defining a load, the directions of the load follow the moving coordinate axis directions if the **Include geometric nonlinearity** checkbox is selected under the **Study settings** section of the current study step. See Coordinate Systems and in the *COMSOL Multiphysics Reference Manual*.

VISUALIZATION

If you have switched on the physics symbols (see Displaying Physics Symbols in the Graphics Window — An Example in the *COMSOL Multiphysics Reference Manual)*, then an applied load is indicated by a symbol together with a coordinate system indicator displaying the definition directions for the load. The actual direction or magnitude of the load you enter is not, however, reflected by the symbol. As a load in COMSOL Multiphysics can be a function of parameters, variables, the solution, or results from other physics interfaces, it is not possible to display it with only the information available in the individual load feature.

Once you have turned on the physics symbols for a certain physics interface, you can fine-tune the display. Every feature which has associated physics symbols will now have a checkbox **Show physics symbols**, by which you can control the display of the symbols for that specific feature.

Loads are among the results for which result templates are generated, so you will always have access to a visual feedback of the loads after the solution. How to work with the default load plots is described in detail in the Plotting Applied Loads section.

Sometimes, especially if you have entered complicated load expressions in a large model, it is important to inspect the load distribution before you run the analysis. You will then need to generate a dataset and the result templates. The fastest way to do that

is to click **Get Initial Value** in the **Study** toolbar. Note that if you change the loads, you will have to delete the dataset and generate it again.

In the *COMSOL Multiphysics Reference Manual*:

- Physics Symbols
- Using Units
- Coordinate Systems
- Plotting Applied Loads

For an example showing how to examine the load distribution, see *Bracket — Static Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_static**. This is also the first example used in the *Introduction to the Structural Mechanics Module*.

Load Cases

For a Stationary or Frequency Domain study, you can define load cases and constraint cases. Any load or constraint can be assigned to a load or constraint group, and then be used conditionally.

For most load types, the load case acts as a simple multiplier, but some cases need a more detailed discussion:

- A **Prescribed Displacement** or **Prescribed Displacement/Rotation** node can be assigned both a constraint group and a load group. You can use the constraint group to switch on and off the whole constraint. The load group acts as a multiplier to any nonzero prescribed values of displacement and rotations, and will not have any effect unless the constraint is active.
- When a load case multiplier is used for **Thermal Expansion**, the multiplier is applied not to the actual temperature, but to the difference between the temperature and the strain free reference temperature. The temperature difference, and thus the thermal strain, is proportional to the load case multiplier.
- The **Thermal Expansion** nodes are exclusive (only the last one given gives a contribution for a certain domain), but you can switch between different **Thermal Expansion** nodes by assigning them to different load cases. Note that the features become contributing when present in the same load case.

- **Hygroscopic Swelling** behaves analogously to **Thermal Expansion**.
- For **External Stress**, the given stress values are multiplied by the load case multiplier.
- A **Spring Foundation** or **Thin Elastic Layer** node can be assigned a constraint group, which you can use to switch the whole boundary condition on and off. If there is also a **Predeformation** subnode, then you can assign a load group to that subnode. The prescribed predeformation is then multiplied by the load case multiplier. **Predeformation** nodes are exclusive, you cannot switch between them by assigning them to different load cases.
- A **Rigid Connector** can be assigned both a constraint group and a load group. You can use the constraint group to switch off the prescribed displacements and rotations. The load group acts as a multiplier to nonzero prescribed values of displacement and rotations.
- If you have added **Applied Force** or **Applied Moment** nodes under a **Rigid Connector**, you can assign individual load groups to these nodes.
- If you have added **Applied Force** or **Applied Moment** nodes under a **Rigid Material**, you can assign individual load groups to these nodes.

Any expression that acts purely as a load, that is, contributes only to the right-hand side of the system of equations, can be part of the load case handling. This is true even if, for example, the corresponding feature does not have a setting for load cases, or if it is a contribution you have created using equation based modeling.

To do this, you can modify any such expression `expr` to `if(group.<lgName>,group.<lgName>*expr)`. Here, `<lgName>` is the parameter name you have chosen in the settings for the load group.

In most cases, the expression `group.<lgName>*expr` is sufficient. The only reason to use the longer version with the if( ) statement if you want to avoid that `expr` is evaluated for load cases in which it is not used.

For example, if you have a boundary load which partially is always active, and partially is conditional, you can write `20[MPa]+group.lg1*10[MPa]` in the input field for the pressure.

In particular, this approach is useful for features that override each other, like Thermal Expansion, since you can then accommodate several load cases in a single node in the Model Builder.

In the *COMSOL Multiphysics Reference Manual*:

- About Load Cases
- Defining Load Groups and Constraint Groups
- Load Group
- Constraint Group

For an example about how to set up expressions for controlling position and distribution of loads using load cases, see *Tapered Cantilever with Two Load Cases*: Application Library path **COMSOL_Multiphysics/Structural_Mechanics/tapered_cantilever**.

Singular Loads

In reality, loads always act on a finite area or over a volume. However, in a model loads are sometimes defined on points or edges, which leads to a singularity. The reason for this is that points and lines have no area, so the stress becomes infinite. Because of the stress singularity, there are high stress values in the area surrounding the applied load. The size of this area and the magnitude of the stress depend on both the mesh and the material properties. The stress distribution at locations far from these singularities is unaffected according a to a well-known principle in solid mechanics, Saint-Venant's principle. It states that for an elastic body, statically equivalent systems of forces produce the same stresses in the body, except in the immediate region where the loads are applied.

Figure 2-15 shows a plate with a hole in plane stress loaded with a distributed load and a point load of the same magnitude. The mesh consists of triangular elements with quadratic shape functions. The high stress around the point load is dissipated within the length of a few elements for both mesh cases. The stresses in the middle of the plate and around the hole agree for the distributed load and the point load. The problem is that due to the high stress around the singular load it is easy to overlook the high stress region around the hole. When the point load is applied, the range must be manually set for the stress plot to get the same visual feedback of the high stress region around the hole in the two cases. This is because the default plot settings automatically set the range based on the extreme values of the expression that is plotted.

Despite these findings it is good modeling practice to avoid singular loads because it is difficult to estimate the size of the singular region. In the Structural Mechanics

Module, it is possible to define loads on all boundary types. However, avoid singular loads altogether with elastoplastic or creep materials.

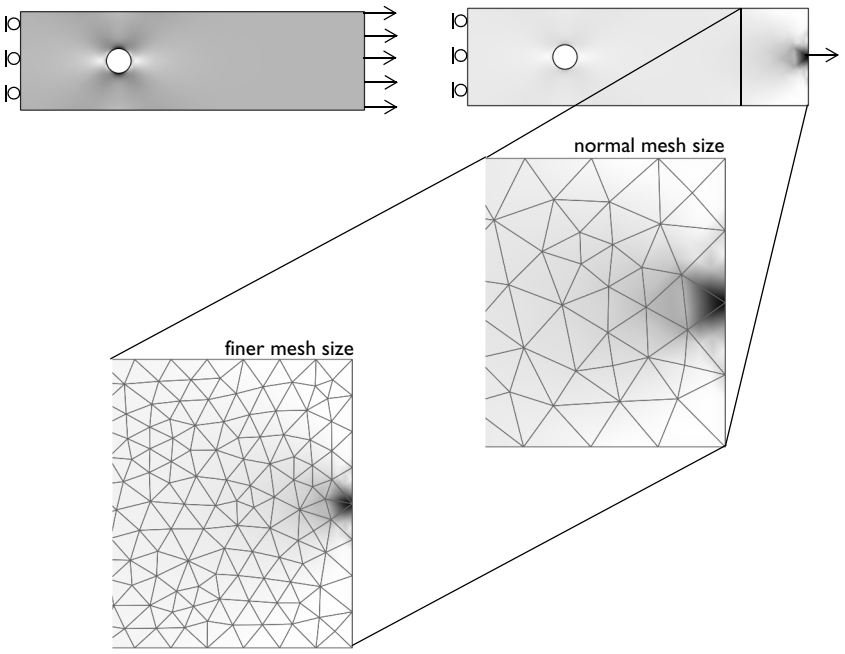


Figure 2-15: A plate with a hole subject to a distributed load (left) and a point load (right).

For more details about singularities, see also

www.comsol.com/blogs/singularities-in-finite-element-models-dealing-with-red-spots

www.comsol.com/blogs/applying-and-interpreting-saint-venants-principle

Moments in the Solid Mechanics Interface

The Solid Mechanics interface, as opposed to the Beam, Pipe Mechanics, Plate, and Shell interfaces, does not have *rotational degrees of freedom*. This makes the direct specification of moment loads somewhat more complicated. To specify moments, you can attach a Rigid Connector to the loaded area. The rigid connector has rotational degrees of freedom, and it is possible to apply moments directly. For load application, the flexible formulation of the rigid connector is particularly useful, since it avoids artificial stiffening of the boundary where the load is applied.

Pressure

A pressure is a load acting toward the normal of a face of the structure. If there are large deformations in the model and the **Include geometric nonlinearity** checkbox is selected under the **Study settings** section of the current study step, the pressure acts as a follower load. The pressure is then defined with respect to the geometry and, as the geometry deforms locally, the orientation of the load changes. The size of the loaded area can also change as an effect of straining.

Acceleration Loads

Within the structural mechanics interfaces, you will find four different types of loads to describe acceleration loads:

- **Gravity**
- **Base Excitation**
- **Rotating Frame**
- **Linearly Accelerated Frame**

The two first nodes have are of a global type. They will be applied to all features in the physics interface, and cannot have a spatial variation.

The **Rotating Frame** and **Linearly Accelerated Frame** nodes have a domain selection. They can be applied selectively, and also depend on the coordinates. When applied to a set of selected domains, such loads are applied also to lower dimensional objects, for example, a point mass or an added mass on an edge.

All acceleration loads share the property that they are not applied to mass contributions that belong to global features such as rigid connectors. There, you must add loads explicitly.

When a structure is subjected to an acceleration which is applied to all its support points, then it often more convenient to replace the support acceleration by an acceleration force, using a **Base Excitation** node. In particular, this is necessary when using modal based dynamic analyses, in which nonzero prescribed displacements cannot be used. Changing to a frame acceleration load does not affect the distribution of forces and stresses. It will however imply that displacements, velocities, and accelerations are measured relative to the supports, and not relative to a room fixed coordinate system. The **Base Excitation** node does however also define absolute acceleration variables, which are what would be measured by an accelerometer.

Temperature Loads — Thermal Expansion

When performing thermal expansion analysis, temperature loads are specified by entering a temperature and a reference temperature in a thermal expansion subnode which is available from the context menu (right-click the parent node, a Linear Elastic Material node, for example) or from the **Physics** toolbar, **Attributes** menu. Enter a constant temperature or an analytic expression that can depend on the coordinates or dependent variables. For beams, plates, and shells it is also possible to specify bending temperature loads. More details are available in the descriptions for each physics interface.

When a separate physics interface is used to model heat transfer in the material, the entry for the temperature is the dependent variable for the temperature from that physics interface, typically T. In most cases, possible temperature variables from other physics interfaces can be directly selected from a list.

- For more information about how to couple heat transfer analysis with structural mechanics analysis, see Thermal–Structure Interaction. This module also includes The Thermal Stress, Solid Interface.
- For a detailed discussion about thermal effects in structural mechanics models, see Thermally Coupled Problems.

Hygroscopic Swelling

Some materials have the capability to absorb significant amounts of moisture through diffusion processes. Changes in the moisture content may then cause volume changes.

To include the effects of hygroscopic swelling, the **Hygroscopic Swelling** subnode is available from the context menu (right-click the parent node, Linear Elastic Material

node, for example) or from the **Physics** toolbar, **Attributes** menu. Enter a constant concentration or an analytic expression that can depend on the coordinates or dependent variables. For beams, plates, and shells it is also possible to specify bending swelling loads caused by concentration gradient in the transverse direction. More details are available in the descriptions for each physics interface.

When a separate physics interface is used to model the moisture diffusion in the material, the entry for the concentration is the dependent variable for the concentration from that physics interface, typically c. In most cases, possible concentration variables from other physics interfaces can be directly selected from a list.

The diffusion of the moisture into the material also adds to the mass density. You can choose to automatically include this effect in a dynamic analysis, and also in mass proportional loads, such as gravity and rotating frame loads.

Total Loads

You can specify a load either as a distributed load per reference or deformed unit length, area, or volume; or as a total force to be distributed on a boundary. In the case of a total load, the applied distributed load is the given load divided by the area (or length, or volume) on which its acts. Thus, entering a total load is usually only meaningful when its orientation is given by a Cartesian coordinate system.

Loads with a Spatial Variation

Since you can write any type of expression in an input text field, it is easy to describe loads having a spatial variation.

HYDROSTATIC LOAD

Hydrostatic loading is a common special case of spatial variation. In this case, there is often a fluid surface, above which there is no load. Such a load you can describe with an expression like `if(Z<ZSurf,rhoFluid*g_const*(ZSurf-Z),0)`. Here, `ZSurf` and `rhoFluid` are assumed to be parameters containing the Z-coordinate of the fluid surface and the mass density of the fluid respectively.

LOADS WITH A FAST SPATIAL VARIATION

If a load has a spatial variation which is fast relative to the element size, you may need to increase the accuracy of the numerical integration used to compute the load contribution. As a default, a load which varies no faster than the polynomial order of

the displacement shape functions can be integrated exactly. To change the integration order, enable Equation View. In the **Equation View** node under the current load node in the Model Builder tree, you can then increase the integration order.

The local stress state within the loaded element is still limited by what can be described by the shape functions, but the total load applied on the structure will be more accurate it you increase the integration order.

TRAVELING LOADS

Loads that are moving along the structure with time can be modeled using an expression `X-v*t`, where `v` is the velocity of the load. The mesh independent point load of the type **Point Load, Free** is particularly well suited for this type of modeling. If a distributed load is modeled using this approach, it is often necessary to increase the integration order as discussed in the previous section, since the load patch will typically cover partial element faces.

Variables for Loads

Each node in which a load is given, such as **Boundary Load** or **Point Load**, creates a number of variables which you have access to, for example during postprocessing. These variables have standardized names. The names are constructed using the following pattern: `<phys>.<loadTag>_<loadType>_<geom><dir>`, for example `solid.bndl1.F_Ax`.

TABLE 2-8: THE LOAD VARIABLE NAMING SCHEME.

	DESCRIPTION	EXAMPLES
`<phys>`	Physics interface tag	`solid`, `truss2`
`<loadTag>`	Tag of the load feature	`bndl1`, `el2`
`<loadType>`	Force or moment	`F` (force) `M` (moment)
`<geom>`	The type of object to which the load is applied	`P` (point) `L` (line) `A` (area) `V` (volume)
`<dir>`	Orientation	`x`, `y`, `z`, `r` (vector components) `_Mag` (magnitude of the vector)

For loads which are do not have a geometrical selection, such as loads on **Rigid Connector** and **Rigid Material**, the `_<geom>` part of the name is omitted. As an example, a load created through an **Applied Force** node under a **Rigid Connector**, is contained in a variable such as `solid.rig2.rf4.Fy`.

Defining Constraints

Defining the proper *constraints* for structural mechanics models is just as important as defining the loads as together they make up the model boundary conditions. There are many useful predefined physics features to define the constraints or to create user-defined expressions that define constraints.

In this section:

- Rigid Body Motion
- Orientation
- Prescribed Displacements, Velocities, and Accelerations
- Symmetry Constraints
- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions
- Kinematic Constraints
- Rotational Joints
- Attachments

Rigid Body Motion

In most cases, a structure must have a set of constraints which is sufficient to suppress any rigid body motions. A stationary problem is solvable only if the structure is sufficiently constrained. There must not be any possible rigid body modes. Thus, no stress-free deformation states are allowed. In a dynamic analysis, rigid body motions are admissible. The inertial forces will then balance the external forces.

A special case, in which it is valid to perform a static analysis on an unconstrained structure is when it is subjected to constant acceleration. This can be achieved through a special analysis type called *inertia relief analysis*.

The number of possible rigid body modes for different geometrical dimensions is shown in the table below.

TABLE 2-9: NUMBER OF POSSIBLE RIGID BODY MODES.

DIMENSION	NUMBER OF RIGID BODY MODES
3D	6 (3 translations + 3 rotations)
2D axisymmetric	1 (Z-direction translation)

TABLE 2-9: NUMBER OF POSSIBLE RIGID BODY MODES.

DIMENSION	NUMBER OF RIGID BODY MODES
2D (solid, beam, truss)	3 (2 translations + 1 rotation)
2D (plate)	3 (1 translation + 2 rotations)

If the model is under constrained, you may encounter the following problems:

- The solver reports that the stiffness matrix is singular.
- The solver reports that the stiffness matrix is ill-conditioned. Theoretically, the matrix is singular for a structure with rigid body modes, but because of the roundoff errors during the solution this is not exactly determined.
- A nonlinear analysis fails to converge.
- An iterative linear equation solver fails to converge.
- You get a solution with an extremely large displacement, orders of magnitude larger than what is expected.

For a single body, it is seldom difficult to see whether it is fully constrained or not, but for a more complex assembly, including several physics interfaces, or advanced couplings and boundary conditions, it may not be trivial. If you suspect that rigid body modes are a problem in your model, you can run an eigenfrequency analysis, and check for modes with zero eigenfrequency as described in Eigenfrequency Analysis.

If there are no constraints which are dictated by the physical boundary conditions of the structure, you can use the Rigid Motion Suppression feature to automatically remove the rigid body motions. When you do this, the assumption is that the external loads are in equilibrium. If not, reaction forces and stress concentrations will appear at seemingly arbitrary points where the automatic constraints were placed.

As an alternative to applying constraints, you can also add elastic supports through a Spring Foundation node to suppress rigid body motion.

Rigid Motion Suppression in the *Structural Mechanics Theory* chapter.

Orientation

You can specify constraints in global as well as in any previously defined local coordinate system.

	Coordinate Systems in the *COMSOL Multiphysics Reference Manual*

Prescribed Displacements, Velocities, and Accelerations

The most fundamental constraint is the prescribed displacement, where the individual components of displacement or rotation can be prescribed to zero or nonzero values for points, edges, boundaries, or domains.

For dynamic analysis, you can also directly prescribe the velocity or acceleration. The conditions for prescribing displacements, velocities, or accelerations are mutually exclusive for the same geometrical object since they prescribe the same degree of freedom.

FREQUENCY DOMAIN

In frequency domain, a prescribed velocity v_p or prescribed acceleration a_p can be directly interpreted as a prescribed displacement u_p:

$$u_p = \frac{v_p}{i\omega}$$

$$u_p = \frac{-a_p}{\omega^2}$$

where ω is the angular frequency.

	A prescribed velocity with zero phase is assumed to have its peak at the reference phase. As an effect, the corresponding peak displacement is shifted by 90°. Similarly, a positive prescribed acceleration with zero phase corresponds to a negative value of the displacement. These definitions are particularly important if you mix prescribed velocity or acceleration conditions with other boundary conditions.

TIME DOMAIN

In the case of a time-dependent analysis, the prescribed displacement is obtained as

$$u_{\mathrm{p}}(t) = u_0(t_0) + \int_{t_0}^{t} v_{\mathrm{p}}(\tau) d\tau$$

or

$$u_{\mathrm{p}}(t) = u_0(t_0) + \int_{t_0}^{t} \left(v_0(t_0) + \int_{t_0}^{\tau} a_{\mathrm{p}}(\zeta) d\zeta \right) d\tau$$

where u_0 and v_0 are is given by the initial conditions. It is not possible to set explicit initial conditions, but if initial values are taken from a previous study, they will be respected. In order to compute the integrals, u_{p} is introduced as a separate degree of freedom which is solved for by adding an extra ODE.

As prescribing the velocity or acceleration in time domain comes with an extra cost, you should always consider using a prescribed displacement instead. As long as the time history of the velocity or acceleration is a known a priori and does not depend on the solution itself, this is possible.

- When the velocity or acceleration has a simple time dependence, you can integrate it analytically one or two times to obtain the displacement, and directly prescribe the displacement instead.
- When you have complicated known velocity or acceleration histories, for example from measurements, you can use the `integrate()` operator. In this case, you enter the prescribed displacement as `integrate(my_data(tau),tau,0,t)`. Here `my_data` is the measured data as function of time, and `tau` is a dummy integration variable

When a local coordinate system is used for prescribing a prescribed velocity or acceleration, the axis directions must be fixed in space. As an example, you cannot use a Boundary System rotating with the deformation.

STATIONARY ANALYSIS

In a stationary analysis, the prescribed velocity and acceleration nodes can have two different behaviors. As a default, they are ignored, but you can also select that the degrees of freedom having a prescribed velocity or acceleration in a dynamic analysis should be constrained to zero in a static analysis.

Symmetry Constraints

In many cases symmetry of the geometry and loads can be used to your advantage in modeling. Symmetries can often greatly reduce the size of a model and hence reduce the memory requirements and solution time. When a structure exhibits axial symmetry, use the axisymmetric physics interfaces. A solid that is generated by rotating a planar shape about an axis is said to have *axial symmetry*. In order to make use of the axisymmetric physics interfaces, all loads and constraints must also be the same around the circumference.

For other types of symmetry, use the predefined symmetry and antisymmetry constraints. This means that no expressions need to be entered — instead just add the type of constraint to apply to the model.

	Physics Interface Axial Symmetry Node in the *COMSOL Multiphysics Reference Manual*

If the geometry exhibits two symmetry planes (Figure 2-16), model a quarter of the geometry by using the *Symmetry* node for the two selected surfaces.

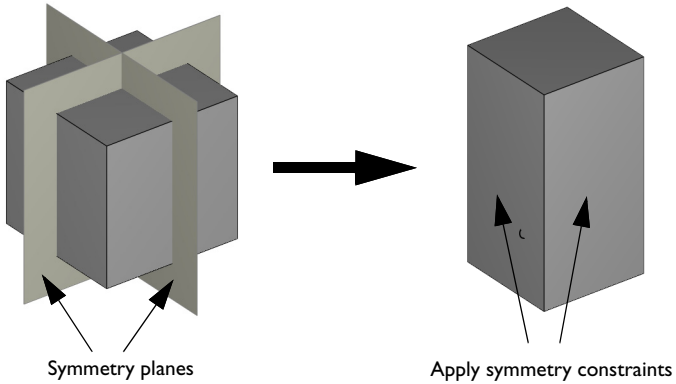


Figure 2-16: If the geometry exhibits two symmetry planes, model a quarter of the geometry by using the Symmetry feature for the two selected surfaces.

	Both geometric symmetry and loads are important when selecting the correct constraints for a model.

!	In an eigenfrequency or buckling analysis, the eigenmodes might be nonsymmetric even if the structure is symmetric.

Figure 2-17 shows symmetric and antisymmetric loading of a symmetric geometry. When modeling half of the geometry, the correct constraint for the face at the middle of the object would be Antisymmetry in the case of antisymmetric loading and Symmetry in the case of symmetric loading of the object.

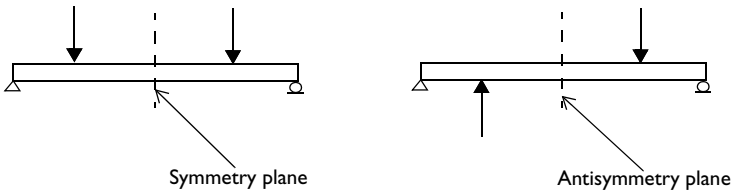


Figure 2-17: Symmetry plane (left) and antisymmetry plane (right).

SYMMETRY IN 2D AXISYMMETRY

In an axisymmetric model, the only possible symmetry is when the symmetry plane is normal to the *Z*-axis. For models in 2d axisymmetry, the **Symmetry Plane** node is used for prescribing this type of symmetry.

Antisymmetry cannot exist in this case.

TRANSLATION OF THE SYMMETRY PLANE

In some situations, you may want to use a symmetry condition, in which the symmetry boundary actually can move along its normal. This may for example the case when you use symmetry conditions to terminate your modeled region even though the situation is not truly symmetric. The best approximation may then be that the boundary remains planar, but that there is no resultant reaction force from the boundary condition.

You can modify the symmetry condition, so that it can translate in various ways by using the controls in the **Normal Direction Condition** section of the settings for the **Symmetry** constraint. You can model the following cases:

- Reaction force free translation.
- Prescribed total force acting on the constrained part.
- The displacement in the normal direction is prescribed.

Note that allowing translation in the symmetry constraint is only meaningful if the geometry selection corresponds to a single symmetry plane.

For an example showing how to force a boundary to remain plane, but still allow it to translate in its normal direction using this special version of **Symmetry**, see *Thermomechanical Analysis of a Surface-Mounted Resistor*: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/surface_resistor**.

Symmetry Condition with Translation in the *Structural Mechanics Theory* chapter.

Elemental and Nodal Constraints

For most constraints, you can select between using elemental and nodal constraints. To do this, select **Advanced Physics Options**, so that the **Constraint Settings** section is displayed.

When using nodal constraints, one constraint is generated for each node within the selection a certain constraint feature. With elemental constraints, the number of constraints added at a node equals the number of elements connected to that node. This means that if some values used in the constraints differ between the elements, then different constraints will be generated by the elemental method, whereas with the nodal method an average is computed at the node before adding the constraint.

When several constraints are present at a node, the internal constraint elimination algorithm is responsible for reducing them to a minimum unique set. Using elemental constraints will clearly put an extra burden on this algorithm, so whenever possible you should use nodal constraints.

The two different options exist, since under some circumstances the actual constraints can differ between the two methods. Consider for example a symmetry constraint, where the displacement in the direction normal to the boundary is constrained by the equation

$$\mathbf{u} \cdot \mathbf{n} = 0$$

where $\mathbf{n}$ is the unit normal vector.

If there are several intersecting symmetry planes, like in Figure 2-16, using nodal constraints could cause a problem:

- If both boundaries are selected in the same **Symmetry** node, then only a single constraint is applied for each node along the common edge, while you actually want constraints along the normals of both planes. The normal used would be pointing somewhere between the two planes, since a nodal constraint uses averaging of the values from the adjacent elements.
- If two **Symmetry** nodes are used, so that the selection in any one of them only contains boundaries without a normal direction discontinuity, the intended constraints are added. On the common edge, there will be two contributions, one from each **Symmetry** node, and each using the normal direction of its boundary. If you want to use nodal constraints, you must set up your model in this way if the constraints are orientation dependent.

Elemental constraints, on the other hand, can cause problems if the constraints added by adjacent elements are not exactly the same. This could for example happen if the normal orientation differs between neighboring elements. In such a case, a boundary could behave as if it were fixed when a **Symmetry**, **Antisymmetry**, or **Roller** constraint is applied. Such a situation could occur when the component consists of an imported mesh, so that no underlying geometry exists.

The default type of the constraint, nodal or elemental, differs between different constraint features. A nodal formulation is the default whenever it is considered safe, like for a **Fixed Constraint**. Whenever the constraint can have a dependency on the surface orientation, the default value is elemental.

See also Constraint Settings in the *COMSOL Multiphysics Reference Manual.*

Suppressing Constraints on Lower Dimensions

Sometimes, boundary conditions on two adjacent objects can come into conflict on a shared object.

For most constraints in the structural mechanics interfaces, you have the possibility to select that certain objects of lower dimensions should be excluded from the main selection. To do this, you must first select **Advanced Physics Options**. In the settings for a constraint, like for example **Prescribed Displacement**, new sections named **Excluded**

Surfaces, **Excluded Edges**, and **Excluded Points** will then appear. In these sections, you can select geometrical objects which should be excluded from the main selection when the constraint is applied.

In the structural mechanics interfaces, there are many types of complex constraints, and sometimes you may get conflicts or duplicates which makes the model either overconstrained, or problematic for the automatic constraint elimination algorithm. If you are aware of such situations, it is good practice to remove one of the potentially conflicting constraints. One example of such a situation is when you have a **Solid–Shell Connection** meeting a symmetry plane, as shown in Figure 2-18.

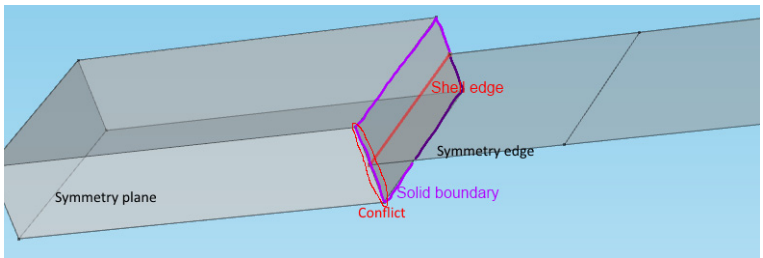


Figure 2-18: Example of potentially conflicting constraints.

Here you would add a **Symmetry** condition on a boundary in the Solid Mechanics interface, as well as a **Symmetry** condition on an edge in the Shell interface. But at the same time, the displacements on whole boundary where the solid meets the shell are controlled by shell degrees of freedom as an effect of the **Solid–Shell Connection**. As a result, on the edge marked with *Conflict* in the sketch, the displacements will be controlled both by the symmetry condition is Solid Mechanics, and implicitly through the coupling, by the symmetry condition in the Shell interface. Particularly if the geometry is curved, there is a risk that these constraints are not identical from a numerical point of view. In this case, excluding the conflicting edge from the selected boundary in the Solid Mechanics interface will make the behavior unique and fully predictable.

Another example where constraints will come in conflict is if you want to constrain the displacement on parts of the geometry using weak constraints, while keeping the default pointwise constraints on other parts. If the same mesh vertex has both types of

constraints, the solution will fail, so you must exclude any common geometrical objects from the selection in one of the constraints.

See also Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

Kinematic Constraints

Kinematic constraints are equations that control the motion of solids, faces, edges, or points. Add a Prescribed Displacement constraint to enter expressions for constraints. You can define the equations using predefined coordinate systems as well as custom coordinate systems. Special constraints, for instance to keep an edge of body straight or to make a boundary rotate, require such constraint equations.

In the 3D and 2D Solid Mechanics interfaces and in the Shell and Beam interface there is a special constraint called a Rigid Connector. A rigid connector is applied to one or more boundaries, edges, or points and force them to behave as connected to a common rigid body. The rigid connector can be given prescribed displacements and rotations and thus simplifies the realization of some constraints.

Rotational Joints

Joints between elements in The Truss Interface are automatically rotational joints because the truss elements have no rotational degrees of freedom. For beams, however, the rotational degrees of freedom are by default coupled between elements. To create a rotational joint between two beam elements, add one additional Beam interface to a geometry. Make sure that it is only active for the edge that includes the point where the joint is positioned and that no other physics interface is active here. Couple the translational degrees of freedom and leave the rotational degrees of freedom uncoupled at the joint.

Attachments

An Attachment is a set of boundaries, edges, or points on a flexible or rigid component used to connect it to another flexible or rigid component through a joint or spring. An

attachment can be defined on the boundaries of a solid element, on the edges of a shell element, and on the points of a beam element.

When an attachment is connected to a flexible body, you can use two different formulations: rigid or flexible.

In the rigid attachment formulation, all selected boundaries or edges behave as if they were connected to a common rigid body. This may cause an unrealistic stiffening and local stress concentrations.

In the flexible version, the boundaries are allowed to deform, and the rigid body constraints are enforced only in an average sense.

The attachment formulation is similar to that of a rigid connector. In the rigid case, the only degrees of freedom needed to represent this assembly are the ones needed to describe the movement of a rigid body. In 2D this is just two in-plane translations, and the rotation around the out-of-plane axis.

In 3D the situation is more complex. Six degrees of freedom are necessary, usually selected as three translations and three parameters for the rotation. For finite rotations any choice of three rotation parameters is singular at some specific set of angles. For this reason, a four-parameter quaternion representation is used.

Some extra degrees of freedom are added for each attachment where the flexible formulation is used.

When an attachment is defined on a rigid component, it does not create any degrees of freedom of its own and directly picks the degrees of freedom of the rigid component.

The formulation of the attachment feature is same as the rigid connector formulation, which is discussed in Rigid Connector in the *Structural Mechanics Module User's Guide*.

Some useful information about the attachment feature:

- The attachment can be defined either on a flexible or on a rigid component. It is not possible to select the boundaries from both types of components in a single attachment. The reason is the different formulation of attachment for flexible or rigid components.

- The attachment's center of rotation is the centroid of its selected boundaries, edges, or points. In a joint, it is possible to select the attachment center of rotation as the center of the joint.
- If the attachment is connected to a flexible domain, the forces and moments on an attachment with a boundary selection are computed by summing the reaction forces on the selected boundaries. There is an option to use these forces and moments to evaluate the joint forces.
- For an attachment in the Shell interface, the selection consists of edges. The forces and moments on an attachment are then computed by summing the reaction forces and reaction moments on the selected edges. There is an option to use these forces and moments to evaluate the joint forces.
- For an attachment in the Beam interface, the selection consists of points. The forces and moments on an attachment are then computed by summing the reaction forces and reaction moments on the selected points. There is an option to use these forces and moments to evaluate the joint forces.
- When an attachment is defined in the Shell or Beam interface, the normal force cannot be defined through **Attachment reaction forces** in the Friction subnode of a joint.
- In a joint you select attachments to establish its connection with flexible, or optionally, rigid bodies. This is accomplished by setting up a relation between the source attachment and destination attachment degrees of freedom. Thus, the motion of an attachment is governed by the joints where this attachment is used.
- An attachment which is not referenced by any joint acts as an unconstrained rigid connector.

- The Multibody Dynamics Interface
- The Solid Mechanics Interface, The Shell and Plate Interfaces, and The Beam Interface in the *Structural Mechanics Module User's Guide*.

Calculating Reaction Forces

There are different ways to evaluate reaction forces and these are discussed in this section.

- Using Predefined Variables to Evaluate Reaction Forces
- Using Weak Constraints to Evaluate Reaction Forces
- Using Surface Traction to Evaluate Reaction Forces
- Evaluating Surface Traction Forces on Interior Boundaries

The following sections describe the merits and costs of these methods.

Using Predefined Variables to Evaluate Reaction Forces

The results analysis capabilities include easy access to the reaction forces and moments. They are available as predefined variables. The reaction force variables are available only at the nodes, and not as a continuous field, so they are not suitable for graphic presentation.

To compute the sum of the reaction forces over a region, use **Volume Integration**, **Surface Integration**, or **Line Integration** under **Results> Derived Values** or under an **Evaluation Group** node. The integration method discovers that the reaction forces are discrete values and applies a summation instead of an integration.

If you create an integration operator under **Component>Definitions> Nonlocal Couplings>Integration** to sum reaction forces, you must explicitly set **Method** to **Summation over nodes**.

Reaction forces are computed as the sum of the nodal values over the selected volume, face, or edge. Reaction moments are calculated as the sum of the moment from the reaction forces with respect to a reference point, and any explicit reaction moments (if

there are rotational degrees of freedom). During result evaluation, you can modify the coordinates of the reference point in the **Parameters** section of a result feature.

- Reaction forces are not available for eigenfrequency analysis or when weak constraints are used.
- Reaction force variables are computed where there are constraints, that is Dirichlet conditions like **Fixed Constraint**, or **Prescribed Displacement**. Reaction force variables are found in the `Reactions` folder in the result menus, and have names like `solid.RFx` and `solid.RMz` for an x-directed force and a moment around the z-axis, respectively.
- From the physical point of view, there can be other sources of reaction forces, such as the elastic and viscous forces from a **Spring Foundation**. Such forces are not included in the reaction force variables, but are placed in separate variables. You will find them in the `Spring and damping forces` folder in the result menus.
- Total reaction force variables are available. In these variables, reaction forces, spring forces, and damping forces have been integrated over the whole physics interface. These variables are found in the `Reactions` folder in the result menus, and have names like `solid.RFtotalx` and `solid.RMtotalz` for an x-directed force and a moment around the z-axis, respectively. These variables should thus, for a stationary analysis, be equal to the total applied load.

- Individual reaction force variables per constraint feature are available for the Solid Mechanics and Shell interfaces. In these variables, reaction forces, spring forces, or Lagrange multipliers (in case of weak constraints) are summed over the nodes or integrated over the selection. These variables are available under the `Reactions` folder in the result menus, and have names like `solid.`*<tag>*`.RFsumx` and `solid.`*<tag>*`.RMsumz` for a force in the x-direction and a moment around the z-axis, respectively. Here, *<tag>* is the tag of the constraint node.
- The ordinary reaction forces are not computed for domains having a **Rigid Material**. The reactions to constraints applied to such domains are, however, included in the total reaction force variables.
- If reaction forces are summed independently for two adjacent boundaries, the total sum is not the same as if the reaction forces were summed for both boundaries in one operation. The values of the nodes at the common edge always contain contributions from the elements at both sides of the edge.

Derived Values, Evaluation Groups, and Tables in the *COMSOL Multiphysics Reference Manual*

Using Weak Constraints to Evaluate Reaction Forces

In the **Constraint Settings** section, select **Weak constraints** from the **Constraint** list in order to compute accurate distributed reactions. Extra variables that correspond to the reaction traction distribution are automatically added to the solution components.

With weak constraints activated, COMSOL Multiphysics adds the reaction forces to the solution components. The variables are denoted `x_lm`, where `x` is the name of the constrained degree of freedom (as, for example, `u_lm` and `v_lm`). The extension `lm` stands for *Lagrange multipliers*. It is only possible to evaluate reaction forces on constrained boundaries in the directions of the constraints.

To compute the total reaction force on a boundary, integrate one of the variables `x_lm` using **Volume Integration**, **Surface Integration**, or **Line Integration** under **Derived Values** or under an **Evaluation Group** node.

If the constraint is defined in a local coordinate system, the degrees of freedom for the weak constraint variables are defined along the directions of that system.

Since the reaction force variables are added to the solution components, the number of DOFs for the model increases slightly, depending on the mesh size for the boundaries in question. Boundaries that are adjacent to each other must have the same constraint settings. The reason for this is that adjacent boundaries share a common node.

Using weak constraints affects the structure of the equation system to be solved, and is not suitable for all types of equation solvers.

In the *COMSOL Multiphysics Reference Manual*:

- Weak Constraints
- Constraint Settings
- Derived Values, Evaluation Groups, and Tables
- Symmetric and Nonsymmetric Constraints

Using Surface Traction to Evaluate Reaction Forces

You can obtain values of the reaction forces on constrained boundaries by using boundary integration of the relevant components of the surface traction vector.

For 2D and 1D axisymmetric components, multiply the surface traction by the cross-section thickness, and for 1D components multiply the surface traction by the cross-section area before integrating to calculate the total reaction force.

Two different types of surface traction results can be computed in COMSOL Multiphysics:

The first type, contained in the variables *interface*.Ta*x*, is computed from the stresses. It is always available. Since the surface traction vector is based on computed stress results, this method is less accurate for computing reactions than the other methods.

The second type, contained in the variables *interface*.Trac*x*, which are based on the first Piola–Kirchhoff stress and represent a force per undeformed area. These nominal traction variables are computed with a higher accuracy when the **Compute boundary fluxes** checkbox in the **Discretization** section is selected for the Solid Mechanics interface.

In case of geometric nonlinearity, the two types of traction variables are interpreted differently. The *interface*.Ta*x* variables are based on Cauchy stress, and contains a force per current area. If you integrate them you must use the spatial frame. The *interface*.Trac*x* variables are based on the first Piola–Kirchhoff stress and contains a force per undeformed area. An integration must then be done on the material frame.

Computing Accurate Fluxes in the *COMSOL Multiphysics Reference Manual*

Evaluating Surface Traction Forces on Interior Boundaries

The boundary flux based tractions are computed not only on exterior boundaries, but also on interior boundaries.

On interior boundaries, there are two traction fields: One acting on each of the domains sharing the boundary.

These internal traction fields are contained in the variables *interface*.iTracu*x* and *interface*.iTracd*x*. The letters u and d in the variable names indicate the upside and downside of the boundary, respectively. If you need the value of the total force acting on an internal section through your model, these variables can be integrated in the material frame.

The *interface*.iTracu*x* and *interface*.iTracd*x* variables are computed with higher accuracy when the **Compute boundary fluxes** checkbox is selected in the **Discretization** section of the physics interface.

Computing Accurate Fluxes in the *COMSOL Multiphysics Reference Manual*

Introduction to Material Models

In this section:

- Material Models for Structural Mechanics
- Entering Material Data
- Introduction to Linear Elastic Materials
- Introduction to Viscoelastic Materials
- Mixed Formulation
- Modeling Damage
- About the Material Databases for the Structural Mechanics Module
- Using External Materials

Material Models for Structural Mechanics

The Structural Mechanics Module without any add-on modules provides the Linear Elastic material with Viscoelasticity and Thermal Expansion modeling capabilities. It also provides access to piezoelectric and magnetostrictive materials.

If you have the optional products Nonlinear Structural Materials Module or Geomechanics Module, many other classes of nonlinear materials are also available. These models can be modified and extended, and custom material models can be defined.

You can also add a material model which you have coded yourself and made available as a binary library file using an External Stress–Strain Relation.

- Modeling Piezoelectric Problems
- Modeling Magnetostrictive Materials

In the *COMSOL Multiphysics Reference Manual*:

- Working with External Materials
- External Material

In and you can find an overview of the families of materials, and their applicability in the various structural mechanics interfaces. Many of the material models can be

augmented by effects like thermal expansion, hygroscopic swelling, initial stresses and strains, external stress, and activation.

Combination of Material Models

It is possible to combine many of the effects in an additive manner. The models based on elasticity all have the same structure where

1. An elastic strain is computed by removing all inelastic strains (for example, plastic or thermal strains) from the total strain.
2. An "elastic stress" is computed from the elastic strains.
3. Any additional stresses (for example viscous stresses, or initial stresses) are added to form the total stress.

This concept will give you a great freedom in combining effects. Some such useful combinations are

- Plasticity and thermal expansion
- Plasticity and creep
- Thermal expansion and hygroscopic swelling
- Viscoelasticity and creep

Entering Material Data

For most material data, you have the option to choose between **From material** and **User defined**. The preferred way of supplying the material data, is through the **Materials** node. If you are using data from the Material Library, this is the only option, but also when supplying your own data this will improve clarity of model.

Constitutive matrices, such as the elasticity matrix for an anisotropic material, are in many cases per definition symmetric. Only the upper diagonal of the matrix given as input is used for forming the matrix used, so you need not enter the lower diagonal part.

Introduction to Linear Elastic Materials

Linear elasticity forms the basis for the majority of structural mechanics simulations.

For isotropic linear elasticity, two parameters are enough to describe the material behavior. The number of parameters increases to (at most) 21 for the fully anisotropic

case in 3D. When setting up a model, make sure that the material parameters are defined in agreement with the type of relationship used. If necessary, transform the material data before entering it in the physics interface. For example, for *orthotropic materials* calculate the Poisson's ratio ν_{xy} by

$$\nu_{xy} = \nu_{yx}\frac{E_x}{E_y}$$

	In the theory section Linear Elastic Material

Introduction to Viscoelastic Materials

The generalized Maxwell, standard linear solid (SLS) and Kelvin–Voigt viscoelastic materials are available. All the models are linear, and the corresponding materials can be described as consisting of one or more branches with a spring and a dashpot acting in parallel to a linear elastic material. For each viscoelastic branch, the shear modulus and the relaxation time (or viscosity) are entered.

	• Linear Viscoelasticity • Viscoelasticity

Mixed Formulation

Nearly and fully incompressible materials can cause numerical problems if only displacements are used in the interpolating functions. Small errors in the evaluation of volumetric strain, due to the finite resolution of the discrete model, are exaggerated by the high bulk modulus. This leads to an unstable representation of stresses, and in general to underestimation of the deformation, because spurious volumetric stresses might balance also applied shear and bending loads.

In several material models you will find settings named **Use mixed formulation** or **Compressibility**, by which you can introduce a mixed formulation.

Use a mixed formulation when the material data is such that the deformation is close to being incompressible. For an isotropic elastic material, this happens when Poisson's ratio approaches 0.5.

There are different approaches to assess which combinations of displacement shape function types and auxiliary variable shape function types yield numerically sound and effective elements. In general, the shape function order of the auxiliary variable should be lower than for the displacement field, to avoid locking. However, this is not a sufficient requirement. The inf-sup condition (Ref. 1) can be used to analytically or numerically identify sound mixed element formulations. In general, the outcome of such a test will depend not only on the shape function type combination, but also on the element type. For instance, a sound and effective combination of shape function types for a tetrahedral element is not necessarily suitable for a hexahedral element. COMSOL Multiphysics provides four types of shape functions for the auxiliary pressure or auxiliary volumetric strain variable. The different shape function types for the auxiliary variable are described below. Depending on the particular context, one of these types is implicitly selected using **Automatic**.

Note that in the case of a mixed mesh, you may have to subdivide the domain by element type, to be able to control the shape functions for the auxiliary variable.

DISCONTINUOUS LAGRANGE

If this shape function type is selected, the auxiliary variable shape function is discontinuous across element boundaries, and it is one order lower than the shape function order of the displacements.

CONTINUOUS

If this shape function type is selected, the auxiliary variable shape function is continuous across element boundaries. It will be of the same type (Lagrange or Serendipity) as that of the displacements, but one order lower. A special case is for a linear displacement field, for which a discontinuous, constant, shape function is used.

LINEAR

If this shape function type is selected, the auxiliary variable is regarded as a linearly interpolated field in the element. In the case of an auxiliary pressure, the interpolation in a 3D isoparametric element is

$$p = p_0 + \xi_1(p_1 - p_0) + \xi_2(p_2 - p_0) + \xi_3(p_3 - p_0)$$

where p_0, p_1, p_2, and p_3 are auxiliary pressure coefficients, and ξ_1, ξ_2, and ξ_3 are isoparametric coordinates. Note that the field is linear in the local element coordinates, and that it is not continuous across element boundaries.

	The linear shape function type is not available for layered material features.

CONSTANT

This shape function type represents a constant auxiliary variable in the element, $p = p_0$.

	The mixed formulation is useful not only for linear elastic materials but also for nonlinear elastic materials, elastoplastic materials, hyperelastic materials, and viscoelastic materials.

	Note that some iterative solvers do not work well together with *mixed formulation* because the stiffness matrix becomes indefinite.

	For more details, see Mixed Formulation in the *Structural Mechanics Theory* chapter.

Modeling Damage

Modeling problems with strain localization might turn into unstable behavior and convergence difficulties. The following techniques can be used to help in such situations:

- Use a displacement-controlled loading scheme, since damage is often associated with a reduction in load carrying capacity, see the *Cracking of a Notched Beam* example in the Geomechanics Module and Nonlinear Structural Materials Module Application Libraries.
- Better convergence is often obtained when the **Nonlinear method** is set to Constant (Newton) with a **Damping factor** equal to 1, and a **Jacobian update** on every iteration.

- Reduce the smallest and/or largest allowable step size in an **Auxiliary sweep**, or restrict the time steps in a time-dependent study.
- Introducing a soft spot where a crack is expected can make the localization of strains more stable, see the *Brittle Damage in Uniaxial Tension* example in the Geomechanics Module Application Library and Nonlinear Structural Materials Module Application Libraries. Alternatively, a random spatial distribution of the material parameters could be employed to obtain a more stable solution.

When using the crack band method or no regularization at all, the following steps are recommended:

- The size of the biggest mesh element h should not exceed $2EG_f/\sigma_{ts}^2$, where E is the Young's modulus, G_f is the fracture energy per reference area, and σ_{ts} is the tensile strength. Larger values of h will cause a snap-back of the stress–strain curve at the material point level.
- Use linear shape functions for the displacement field. When using higher-order shape functions, strains may localize in either rows of Gauss points, or entire elements, depending on the stress state and numerical rounding errors.
- If cracks are located on a symmetry plane, the model parameters should be modified so that the amount of dissipated energy is reduced by one half in the elements adjacent to the symmetry plane. This can be achieved, for example, by reducing the fracture energy in that row of elements.

When using the implicit gradient method, the element size should be sufficiently small to resolve damaged zones. The same applies to the **Phase-field damage** model, where it is recommended that size of the mesh elements in the expected crack path follows $h < l_{int}/2$ for a linear displacement field, otherwise $h < l_{int}$.

While it is possible to solve brittle fracture problems with the **Phase-field damage** model by applying a fully coupled strategy, this can often exhibit poor or slow convergence. An alternative and often more stable approach is to use a segregated solution strategy, by splitting the evolution of the crack phase field and the displacement field in two groups. This type of algorithmic operator split can conceptually be summarized as follows for step $n + 1$:

1. *Initialization*. The crack phase field, displacement field and state variables are known at step n.
2. *Update state variables.* Update internal state variables used by the phase field model with values from step n.

3 *Solve for the Crack Phase Field*. Compute the crack phase field variable in a Newton step, with the displacement field frozen at step n.

4 *Solve for the Displacement field*. Compute displacement field variables in a Newton step with the updated crack phase field.

This leads to a single-pass algorithm that is accurate only for sufficiently small parameter or time steps.

The default solver will suggest the above single-pass algorithm for the **Phase-field damage** model when it is feasible to perform the operator split. Cases where this is not possible include when some multiphysics couplings are present in the model and when a segregated contact algorithm has to be used.

An improvement to the method is to add a multipass correction by iterating over steps 3 and 4 in each increment; either until convergence is achieved or for a predefined number of iterations. This type of strategy is demonstrated in the *Brittle Fracture of a Holed Plate* example in the Geomechanics Module and Nonlinear Structural Materials Module Application Libraries, where a multipass algorithm with a maximum of 3 outer iterations is used. Note that although the solution is accepted without requiring convergence of the outer problem, each subgroup locally fulfills the defined convergence criterion. Hence the displacement field can be considered as a converged solution given the current crack phase field.

About the Material Databases for the Structural Mechanics Module

The Structural Mechanics Module includes these material databases: Liquids and Gases, with temperature-dependent fluid dynamic and thermal properties, MEMS, an extended solid materials library with metals, semiconductors, insulators, and polymers common in MEMS devices, and a Piezoelectric database with over 20 common

piezoelectric materials. The materials include temperature-dependent fluid dynamic and thermal properties.

	In the *COMSOL Multiphysics Reference Manual*: • MEMS Material Library • Piezoelectric Materials Library • Liquids and Gases Material Library • Materials

	For an example of the MEMS materials database and Piezoelectric materials database, see *Piezoelectric Shear-Actuated Beam*: Application Library path **Structural_Mechanics_Module/Piezoelectric_Effects/shear_bender**.

Using External Materials

If you need to use a constitutive model which is not available among the built-in material models, it is possible to program it yourself. Such a material function, termed an *external material*, is coded in C. If you already have an existing code in another language like Fortran or C++, it is however possible to write a wrapper function to it.

Before moving to implementing your own material model, there are however two other options to consider:

- Many material models like hyperelasticity, creep, and plasticity have **User defined** as one of the options in addition to the standard models. Any material model which you can describe using built-in variables is most conveniently described here.
- A material model which can partially described in terms of a PDE can often be implemented using one of the mathematical interfaces. Stresses or strains computed in that interface are then injected in an existing material model using the External Stress and External Strain subnodes.

There are two basic types of external material functions: those which completely replace other material definitions in a domain, and those that just compute an inelastic strain contribution to be used as part of an existing material model. The former is referenced from an External Stress–Strain Relation node, whereas the latter is referenced from an External Strain subnode.

During the solution, an external material routine is always called for each Gauss point during evaluation of stiffness matrices and computation of residuals. During result presentation, the external material can be called from any location in the geometry, as requested by for example graphs and point evaluations.

Almost invariably, you need to store state variables in the external material, such as for example plastic strains. The state variables are stored at the Gauss points. If an external material is called at another location, the state variables will be interpolated to that location. This means that the state of the material may not be exactly consistent there, which can lead to some artifacts during result presentation. You can avoid this problem by using the gpeval operator.

In the *COMSOL Multiphysics Reference Manual*:

- Working with External Materials
- External Material

LIBRARY OF UTILITY FUNCTIONS

In order to simplify the task of writing the code for an external material, a library of utility routines is provided. It provides a toolkit for operations common in solid mechanics such as various tensor operations or computing principal stresses and equivalent stresses.

LIST OF UTILITY FUNCTIONS

csext_add: Function that adds two matrices

```
/*
 * Function: csext_add
 * -------------------
 * Description:
 *    Adds two (3 x 3) matrices 'A' and 'B' and stores the
 *    result 'A + B' in 'C'.
 *
 * Arguments:
 *    double A[3][3]
 *    double B[3][3]
 *    double C[3][3] (output)
 *
 * Return value:
 *    void
 */
void csext_add(double A[3][3],double B[3][3], double C[3][3]);
```

csext_det: Function that computes the matrix determinant

```
/*
 * Function: csext_det
 * -------------------
 * Description:
 *    Computes the determinant of a (3 x 3) matrix 'A' and
 *    returns the value.
 *
 * Arguments:
 *    double A[3][3]
 *
 * Return value:
 *    double
 *
 */
double csext_det(double A[]3[3]);
```

csext_dev: Function that computes the deviator of a matrix

```
/*
 * Function: csext_dev
 * -------------------
 * Description:
 *    Computes the deviator of a (3 x 3) matrix 'A'
 *    and stores the value in 'dev'.
 *
 * Arguments:
 *    double A[3][3]
 *    double dev[3][3] (output)
 *
 * Return value:
 *    void
 *
 */
void csext_dev(double A[3][3], double dev[3][3]);
```

csext_dot: Function that computes the inner product of two matrices

```
/*
 * Function: csext_dot
 * -------------------
 * Description:
 *    Computes the inner product of (3 x 3) matrices 'A' and 'B',
 *    and returns 'A : B'.
 *
 * Arguments:
 *    double A[3][3]
```

```
 *    double B[3][3]
 *
 * Return value:
 *    double
 *
 */
double csext_dot(double A[3][3],double B[3][3]);
```

csext_eig: Function that computes the spectral decomposition of a symmetric matrix

```
/*
 * Function: csext_eig
 * -------------------
 * Description:
 *    Computes the eigenvalues and eigenvectors of a
 *    symmetric (3 x 3) matrix 'A'.
 *    The eigenvalues are stored in 'vals', sorted with the
 *    largest value in vals[0].
 *    The eigenvectors are stored column-wise in 'vecs' with
 *    the same ordering as the eigenvalues.
 *    Normal execution returns 0. nonzero means that the
 *    computation failed.
 *
 * Arguments:
 *    double A[3][3]
 *    double vals[3]    (output)
 *    double vecs[3][3] (output)
 *
 * Return value:
 *    int
 *
 */
int csext_eig(double A[3][3], double vals[3], double vecs[3][3]);
```

csext_gl: Function that computes the Green–Lagrange strain tensor

```
/*
 * Function: csext_gl
 * ------------------
 * Description:
 *    Computes the Green–Lagrange strain tensor 'egl' based on
 *    the Right Cauchy–Green deformation tensor 'rcg'.
 *
 * Arguments:
 *    double rcg[3][3]
 *    double egl[3][3] (output)
 *
 * Return value:
 *    void
```

```
 *
 */
void csext_gl(double rcg[3][3],double egl[3][3]);
```

csext_inv: Function that computes the matrix inverse

```
/*
 * Function: csext_inv
 * -------------------
 * Description:
 *    Computes the inverse of a (3 x 3) matrix 'A'.
 *    The inverse, if it exists, is stored in 'inv'.
 *    0 is returned if successful, -1 if 'A' is numerically
 *    singular. The matrix 'A' is considered singular if
 *    abs(det(A))<tol.
 *
 * Arguments:
 *    double A[3][3]
 *    double tol
 *    double inv[3][3] (output)
 *
 * Return value:
 *    int
 *
 */
int csext_inv(double A[3][3], double tol, double inv[3][3]);
```

csext_lcg: Function that computes the Left Cauchy–Green deformation tensor

```
/*
 * Function: csext_lcg
 * -------------------
 * Description:
 *    Computes the Left Cauchy–Green deformation 'lcg' tensor
 *    based on the deformation gradient 'defgrad'.
 *
 *
 * Arguments:
 *    double defgrad[3][3]
 *    double lcg[3][3] (output)
 *
 * Return value:
 *    void
 *
 */
void csext_lcg(double defgrad[3][3], double lcg[3][3]);
```

csext_linsolv: Function to solve a linear system of equations

```
/*
 * Function: csext_linsolv
 * -------------------
 * Description:
 *    Solves a linear system of equations, 'Ax = b', for 'x',
 *    where 'A' is (n x n), and 'b' is (n x 1).
 *    The maximum allowed size of the system is n = 6.
 *
 *    The solution vector 'x' is stored in 'b' as output.
 *    The elements of matrix 'A' are not preserved.
 *
 *    If successful, 0 is returned.
 *    If the solution cannot be determined, -1 is returned.
 *    If n < 1 or n > 6, -2 is returned.
 *
 * Arguments:
 *    double *A
 *    double *b     (input/output)
 *
 * Return value:
 *    int
 *
 */
CSEXTUTILS_SYMBOLS int csext_linsolv(int n, double *A, double *b);
```

csext_lpolar: Function that computes the Left polar decomposition of a matrix

```
/*
 * Function: csext_lpolar
 * -------------------
 * Description:
 *    Computes the Left polar decomposition F = VR,
 *    such that the deformation gradient 'defgrad' is
 *    multiplicatively decomposed into a rotation, 'R',
 *    and a stretch tensor, 'V'.
 *    If the polar decomposition fails, -1 is returned.
 *    If successful, 0 is returned.
 *
 * Arguments:
 *    double defgrad[3][3]
 *    double V[3][3]     (output)
 *    double R[3][3]     (output)
 *
 * Return value:
 *    int
 *
 */
int csext_lpolar(double defgrad[3][3], double V[3][3],
```

```
double R[3][3]);
```

csext_mises: Function that computes the von Mises equivalent stress

```
/*
 * Function: csext_mises
 * ---------------------
 * Description:
 *    Computes and returns the von Mises equivalent stress based
 *    on a stress tensor 'sig'.
 *
 * Arguments:
 *    double sig[3][3]
 *
 * Return value:
 *    double
 *
 */
double csext_mises(double sig[3][3]);
```

csext_mul: Function that multiplies two matrices

```
/*
 * Function: csext_mul
 * -------------------
 * Description:
 *    Multiplies two (3 x 3) matrices 'A' and 'B'.
 *    The result 'AB' is stored in 'C'.
 *
 * Arguments:
 *    double A[3][3]
 *    double B[3][3]
 *    double C[3][3] (output)
 *
 * Return value:
 *    void
 *
 */
void csext_mul(double A[3][3], double B[3][3], double C[3][3]);
```

csext_rcg: Function that computes the Right Cauchy–Green deformation tensor

```
/*
 * Function: csext_rcg
 * -------------------
 * Description:
 *    Computes the Right Cauchy–Green deformation tensor 'rcg'
 *    based on the deformation gradient 'defgrad'.
```

```
 *
 * Arguments:
 *    double defgrad[3][3]
 *    double rcg[3][3]     (output)
 *
 * Return value:
 *    void
 *
 */
void csext_rcg(double defgrad[3][3], double rcg[3][3]);
```

csext_rpolar: Function that computes the Right polar decomposition of a matrix

```
/*
 * Function: csext_rpolar
 * ------------------
 * Description:
 *    Computes the Right polar decomposition F = RU,
 *    such that the deformation gradient 'defgrad' is
 *    multiplicatively decomposed into a rotation, 'R',
 *    and a stretch tensor, 'U'.
 *    If the polar decomposition fails, -1 is returned.
 *    If successful, 0 is returned.
 *
 * Arguments:
 *    double defgrad[3][3]
 *    double R[3][3]     (output)
 *    double U[3][3]     (output)
 *
 * Return value:
 *    int
 *
 */
int csext_rpolar(double defgrad[3][3], double R[3][3],
double U[3][3]);
```

csext_spect: Function that computes a matrix based on its spectral decomposition

```
/*
 * Function: csext_spect
 * ------------------
 * Description:
 *    Computes a symmetric (3 x 3) matrix 'A' based on
 *    its spectral decomposition A=Q*diag*Q^T.
 *    The matrix 'diag' is diagonal and stores the
 *    eigenvalues of 'A'. The matrix 'Q' stores the
 *    eigenvectors (column-wise) of 'A', with the ordering
 *    corresponding to the eigenvalues in 'diag'.
 *    The vector 'd' stores the diagonal elements of 'diag'.
```

```
 *
 * Arguments:
 *    double Q[3][3]
 *    double d[3]
 *    double A[3][3]     (output)
 *
 * Return value:
 *    void
 *
 */
void csext_spect(double Q[3][3], double d[3], double A[3][3]);
```

csext_trace: Function that computes the matrix trace

```
/*
 * Function: csext_trace
 * ---------------------
 * Description:
 *    Computes and returns the trace of a (3 x 3) matrix 'A'.
 *
 * Arguments:
 *    double A[3][3]
 *
 * Return value:
 *    double
 *
 */
double csext_trace(double A[3][3]);
```

csext_transp: Function that computes the matrix transpose

```
/*
 * Function: csext_transp
 * ----------------------
 * Description:
 *    Computes the transpose of a (3 x 3) matrix 'A'.
 *    The result is stored in 'transp'.
 *
 * Arguments:
 *    double A[3][3]
 *    double transp[3][3] (output)
 *
 * Return value:
 *    void
 *
 */
void csext_transp(double A[3][3], double transp[3][3]);
```

csext_add_voigt: Function that adds two matrices

```
/*
 * Function: csext_add_voigt
 * -------------------------
 * Description:
 *    Adds two matrices 'A' and 'B' stored on Voigt form.
 *    The result is stored in 'C'.
 *
 * Arguments:
 *    double A[6]
 *    double B[6]
 *    double C[6] (output)
 *
 * Return value:
 *    void
 */
void csext_add_voigt(double A[6], double B[6], double C[6]);
```

csext_dev_voigt: Function that computes the deviator of a symmetric matrix

```
/*
 * Function: csext_dev_voigt
 * -------------------------
 * Description:
 *    Computes the deviator of a (symmetric) matrix 'A' stored on
Voigt form.
 *    The result is stored in 'dev'.
 *
 * Arguments:
 *    double A[6]
 *    double dev[6] (output)
 *
 * Return value:
 *    void
 */
void csext_dev_voigt(double A[6], double dev[6]);
```

csext_dot_voigt: Function that computes the inner product of symmetric matrices

```
/*
 * Function: csext_dot_voigt
 * -------------------------
 * Description:
 *    Computes and returns the dot product (inner product) of two
 *    symmetric (3 x 3) matrices 'A' and 'B' stored on Voigt form.
 *
 * Arguments:
 *    double A[6]
```

```
 *    double B[6]
 *
 * Return value:
 *    double
 */
double csext_dot_voigt(double A[6], double B[6]);
```

csext_mises_voigt: Function that computes the von Mises equivalent stress

```
/*
 * Function: csext_mises_voigt
 * ---------------------
 * Description:
 *    Computes and returns the von Mises equivalent stress based
 *    on a stress tensor 'sig' on Voigt form.
 *
 * Arguments:
 *    double sig[6]
 *
 * Return value:
 *    double
 *
 */
double csext_mises_voigt(double sig[6]);
```

csext_utils_trace_voigt: Function that computes the matrix trace

```
/*
 * Function: csext_trace_voigt
 * --------------------------
 * Description:
 *    Computes and returns the trace of a symmetric (3 x 3) matrix
 *    'A' stored on Voigt form.
 *
 * Arguments:
 *    double A[6]
 *
 * Return value:
 *    double
 */
double csext_trace_voigt(double A[6]);
```

csext_from_voigt: Function to change from Voigt notation

```
/*
 * Function: csext_from_voigt
 * ------------------
 * Description:
```

```
 *    Converts a symmetric (3 x 3) matrix 'A' stored
 *    on Voigt form to matrix form. The result is stored in 'B'.
 *    If 'def' = 1, the values of last three elements of 'A' are
 *    un-altered when passed into 'B'.
 *    If 'def' = 2, the values of last three elements of 'A' are
 *    halved when passed into 'B'.
 *    Returns -1 if the value of 'def' is invalid, 0 otherwise.
 *
 * Arguments:
 *    double A[6]
 *    int def
 *    double B[3][3] (output)
 *
 * Return value:
 *    int
 *
 */
int csext_from_voigt(double A[6], int def, double B[3][3]);
```

csext_to_voigt: Function to change to Voigt notation

```
/*
 * Function: csext_to_voigt
 * ------------------
 * Description:
 *    Converts a symmetric (3 x 3) matrix 'A' to Voigt form.
 *    The result is stored in 'B'.
 *    If 'def' = 1, the values of the off-diagonal components
 *    of 'A' are unaltered when passed into 'B'.
 *    If 'def' = 2, the values of the off-diagonal components
 *    of 'A' are doubled  when passed into 'B'.
 *    Returns -1 if the value of 'def' is invalid, 0 otherwise.
 *
 * Arguments:
 *    double A[3][3]
 *    int def
 *    double B[6] (output)
 *
 * Return value:
 *    int
 *
 */
int csext_to_voigt(double A[3][3], int def, double B[6]);
```

csext_jac_conv: Function to convert the Jacobian

```
/*
 * Function: csext_jac_conv
 * ------------------------
```

```
 * Description:
 *   Converts a Jacobian from 'dSde' to 'dSdF' using the deformation
 *    gradient 'defgrad', where:
 *     - 'S' is the 2:nd Piola–Kirchhoff stress tensor,
 *     - 'e' is the Green–Lagrange strain tensor,
 *     - 'F' is the deformation gradient.
 *    If 'def' = 1, dSde is defined using tensor shears.
 *    If 'def' = 2, dSde is defined using engineering shears.
 *    Returns -1 if the value of 'def' is invalid.
 *
 * Arguments:
 *    double dSde[6][6]
 *    int def
 *    double defgrad[3][3]
 *    double dSdF[6][9]    (output)
 *
 * Return value:
 *    int
 *
*/
int csext_jac_conv(double dSde[6][6], int def,
double defgrad[3][3], double dSdF[6][9]);
```

csext_unit: Function to define a unit matrix

```
/*
 * Function: csext_unit
 * --------------------
 * Description:
 *    Initializes an (n x n) matrix 'A' to the identity matrix.
 *    Returns -1 if n < 1,
 *    0 otherwise.
 *
 * Arguments:
 *    int n
 *    double *A (output)
 *
 * Return value:
 *    int
 *
 */
int csext_unit(int n, double *A);
```

csext_zero: Function to initialize a matrix

```
/*
 * Function: csext_zero
 * --------------------
 * Description:
```

```
 *    Initializes an (m x n) matrix 'A' to zero.
 *    Returns -1 if m < 1 or n < 1,
 *    0 otherwise.
 *
 * Arguments:
 *    int m
 *    int n
 *    double *A (output)
 *
 * Return value:
 *    void
 *
 */
int csext_zero(int m, int n, double *A);
```

References

1. D. Chapelle and K.J. Bathe, *The Inf-Sup test*, Comp. Struct., Vol. 47, No. 4/5, pp. 537–545, 1993.

Modeling Piezoelectric Problems

In this section:

- About Piezoelectric Materials
- Piezoelectric Coupling
- Create the Piezoelectricity Interface and Define Domains
- Complete Settings of Piezoelectric Materials
- Add Damping and Loss
- Define Material Properties

- Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual.*
- Piezoelectric Coupling
- Piezoelectricity in the theory section

About Piezoelectric Materials

Piezoelectric materials become electrically polarized when strained. From a microscopic perspective, the displacement of atoms within the unit cell (when the solid is deformed) results into electric dipoles within the medium. In certain crystal structures, this combines to give an average macroscopic dipole moment or electric polarization. This effect, known as the *direct piezoelectric effect*, is always accompanied by the *converse piezoelectric effect*, in which the solid becomes strained when placed in an electric field.

Within a piezoelectric, there is a coupling between the strain and the electric field, which is determined by the constitutive relation:

$$\begin{aligned} S &= s_E T + d^T \mathbf{E} \\ \mathbf{D} &= dT + \varepsilon_T \mathbf{E} \end{aligned} \tag{2-5}$$

Here, the naming convention used in piezoelectricity theory is assumed: S is the strain, T is the stress, $\mathbf{E}$ is the electric field, and $\mathbf{D}$ is the electric displacement field. The material parameters s_E, d, and ε_T, correspond to the material compliance, the coupling properties and the permittivity. These quantities are tensors of rank 4, 3, and 2,

respectively. The tensors, however, are highly symmetric for physical reasons, and they can be represented as matrices within an abbreviated subscript notation, which is usually more convenient. In the Piezoelectricity interface, the Voigt notation is used, which is a standard in the literature for piezoelectricity but which differs from the defaults in the Solid Mechanics interface.

	The Piezoelectric Material uses the structural mechanics nomenclature. The strain is named ε (instead of **S**) and the stresses are denoted by either σ or *S* (instead of **T**). This makes the names consistent with those used in the other structural mechanics interfaces.

Equation 2-5 will, using the notation from structural mechanics, then read

$$\begin{aligned} \varepsilon &= s_E\sigma + d^T\mathbf{E} \\ \mathbf{D} &= d\sigma + \varepsilon_0\varepsilon_{rT}\mathbf{E} \end{aligned} \tag{2-6}$$

Equation 2-5 (or Equation 2-6) is known as the *strain–charge form* of the constitutive relations. The equation can be rearranged into the *stress–charge form*, which relates the material stresses to the electric field:

$$\begin{aligned} \sigma &= c_E\varepsilon - e^T\mathbf{E} \\ \mathbf{D} &= e\varepsilon + \varepsilon_0\varepsilon_{rS}\mathbf{E} \end{aligned} \tag{2-7}$$

The material properties, c_E, e, and ε_S are related to s_E, d, and ε_T. It is possible to use either form of the constitutive relations. In addition to Equation 2-5 or Equation 2-7, the equations of solid mechanics and electrostatics must also be solved within the material.

	• Piezoelectric Coupling • Modeling Piezoelectric Problems • Piezoelectricity in the theory section

Piezoelectric Material Orientation

The orientation of a piezoelectric crystal cut is frequently defined by the system introduced by the IRE standard of 1949 (Ref. 8). This standard has undergone a number of subsequent revisions, with the final revision being the IEEE standard of

1987 (Ref. 9). Unfortunately, the 1987 standard contained a number of serious errors and the IEEE subsequently withdrew it. COMSOL therefore adopts the preceding 1978 standard (Ref. 10), which is similar to the 1987 standard, for material property definitions. Most of the material properties in the material library are based on the values given in the book by Auld (Ref. 11), which uses the 1978 IEEE conventions. This is consistent with general practice except in the specific case of quartz, where it is more common to use the 1949 IRE standard to define the material properties. COMSOL therefore provides additional sets of material properties consistent with this standard for the case of quartz. Note that the material properties for quartz are based on Ref. 12, which uses the 1949 IRE standard (the properties are appropriately modified according to the different standards).

The stiffness, compliance, coupling, and dielectric material property matrices are defined with the crystal axes aligned with the local coordinate axes. Note that the signs of several matrix components differ between the 1949 and the 1978 standards (see Table 2-10). In the absence of a user-defined coordinate system, the local system corresponds to the global X, Y, and Z coordinate axes. When an alternative coordinate system is selected this system defines the orientation of the crystal axes. This is the mechanism used in COMSOL to define a particular crystal cut, and typically it is necessary to calculate the appropriate Euler angles for the cut (given the thickness orientation for the wafer). All piezoelectric material properties are defined using the Voigt form of the abbreviated subscript notation, which is universally employed in the literature (this differs from the standard notation used for the Solid Mechanics interface material properties). The material properties are defined in the material frame, so that if the solid rotates during deformation the material properties rotate with the solid. See Modeling Geometric Nonlinearity.

Crystal cuts are usually defined by a mechanism introduced by the IEEE/IRE standards. Both standards use a notation that defines the orientation of a virtual slice (the plate) through the crystal. The crystal axes are denoted X, Y, and Z and the plate, which is usually rectangular, is defined as having sides l, w, and t (length, width, and thickness). Initially the plate is aligned with respect to the crystal axes and then up to three rotations are defined, using a right-handed convention about axes embedded along the l, w, and t sides of the plate. Taking AT cut quartz as an example, the IEEE 1978 standard defines the cut as: (YXl) $-35.25°$. The first two letters in the bracketed expression always refer to the initial orientation of the thickness and the length of the plate. Subsequent bracketed letters then define up to three rotational axes, which move with the plate as it is rotated. Angles of rotation about these axes are specified after the bracketed expression in the order of the letters, using a right-handed convention. For

AT cut quartz only one rotation, about the l axis, is required. This is illustrated in Figure 2-20. Note that within the 1949 IRE Standard AT cut quartz is denoted as: (YXl) +35.25°, since the X-axis is rotated by 180° in this convention and positive angles therefore correspond to the opposite direction of rotation (see Figure 2-19). Table 2-11 summarizes the differences between the standards for different crystal cuts.

When defining the material properties of Quartz, the orientation of the X, Y, and Z axes with respect to the crystal differs between the 1987 IEEE standard and the 1949 IRE standard. Figure 2-19 shows the alignment of the axis for the case of right-handed quartz. A consequence of this is that both the material property matrices and the crystal cuts differ between the two standards. Table 2-10 summarizes the signs for the important matrix elements under the two conventions. Table 2-11 shows the different definitions of the crystal cuts under the two conventions.

TABLE 2-10: SIGNS FOR THE MATERIAL PROPERTIES OF QUARTZ, WITHIN THE TWO STANDARDS COMMONLY EMPLOYED.

	IRE 1949 STANDARD		IEEE 1978 STANDARD	
MATERIAL PROPERTY	RIGHT HANDED QUARTZ	LEFT HANDED QUARTZ	RIGHT HANDED QUARTZ	LEFT HANDED QUARTZ
s_{14}	+	+	-	-
c_{14}	-	-	+	+
d_{11}	-	+	+	-
d_{14}	-	+	-	+
e_{11}	-	+	+	-
e_{14}	+	-	+	-

TABLE 2-11: CRYSTAL CUT DEFINITIONS FOR QUARTZ CUTS WITHIN THE TWO STANDARDS COMMONLY EMPLOYED AND THE CORRESPONDING EULER ANGLES FOR DIFFERENT ORIENTATIONS OF THE CRYSTAL THICKNESS.

STANDARD	REPRESENTATION	AT CUT	BT CUT
IRE 1949	Standard	(YXl) +35.25°	(YXl) –49°
	Y-thickness Euler angles	(ZXZ: 0°,–35.25°,0°)	(ZXZ: 0°,49°,0°)
	Z-thickness Euler angles	(ZXZ: 0°,–125.25°,0°)	(ZXZ: 0°, –41°,0°)
IEEE 1978	Standard	(YXl) –35.25°	(YXl) +49°

TABLE 2-11: CRYSTAL CUT DEFINITIONS FOR QUARTZ CUTS WITHIN THE TWO STANDARDS COMMONLY EMPLOYED AND THE CORRESPONDING EULER ANGLES FOR DIFFERENT ORIENTATIONS OF THE CRYSTAL THICKNESS.

STANDARD	REPRESENTATION	AT CUT	BT CUT
	Y-thickness Euler angles	(*ZXZ*: 0°, 35.25°,0°)	(*ZXZ*: 0°, –49°,0°)
	Z-thickness Euler angles	(*ZXZ*: 0°, –54.75°,0°)	(*ZXZ*: 0°, –139°,0°)

When defining the material orientation, it is necessary to consider the orientation of the plate with respect to the global coordinate system in addition to the orientation of the plate with respect to the crystallographic axes. Consider the example of AT cut quartz in Figure 2-20. The definition of the appropriate local coordinate system depends on the desired final orientation of the plate in the global coordinate system. One way to set up the plate is to orient its normal parallel to the *Y*-axis in the global coordinate system. Figure 2-21 shows how to define the local coordinate system in this case. Figure 2-22 shows how to define the local system such that the plate has its normal parallel to the global *Z*-axis. In both cases it is critical to keep track of the orientation of the local system with respect to the global system, which is defined depending on the desired orientation of the plate in the model.

There are also a number of methods to define the local coordinate system with respect to the global system. Usually, it is most convenient to define the local coordinates with a Rotated System node, which defines three Euler angles according to the *ZXZ* convention (rotation about *Z*, then *X*, then *Z* again). Note that these Euler angles define the local (crystal) axes with respect to the global axes — this is distinct from the approach of defining the cut (global) axes with respect to the crystal (local) axes.

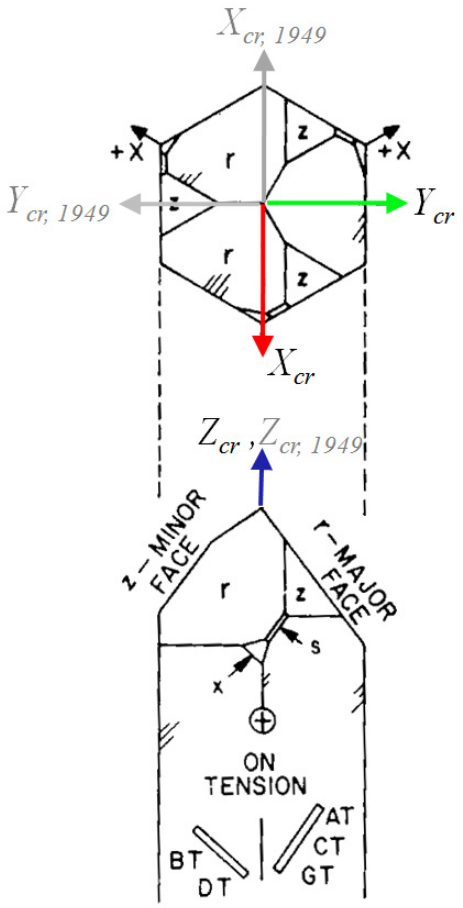


Figure 2-19: Crystallographic axes defined for right-handed quartz in COMSOL and the 1978 IEEE standard (color). The 1949 standard axes are shown for comparison (gray).

Figure 2-19 is reproduced with permission from: *IEEE Std 176-1987 - IEEE Standard on Piezoelectricity*, reprinted with permission from IEEE, 3 Park Avenue, New York, NY 10016-5997 USA, copyright 1987, by IEEE. This figure must not be reprinted or further distributed without prior written permission from the IEEE.

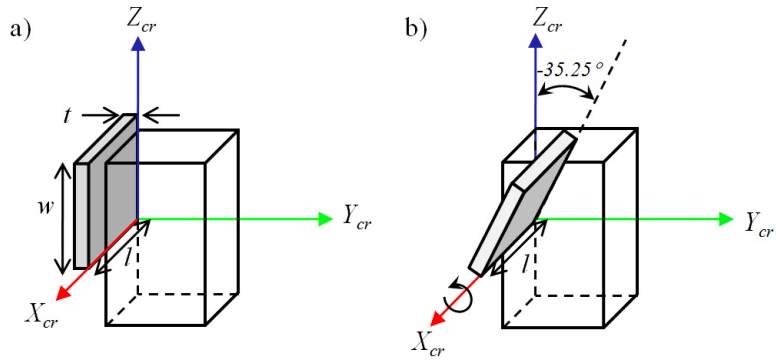


Figure 2-20: Definition of the AT cut of quartz within the IEEE 1978 standard. The AT cut is defined as: (YXl) –35.25°. The first two bracketed letters specify the initial orientation of the plate, with the thickness direction, t, along the crystal Y-axis and the length direction, l, along the X-axis. Then up to three rotations about axes that move with the plate are specified by the corresponding bracketed letters and the subsequent angles. In this case only one rotation is required about the l-axis, of –35.25° (in a right-handed sense).

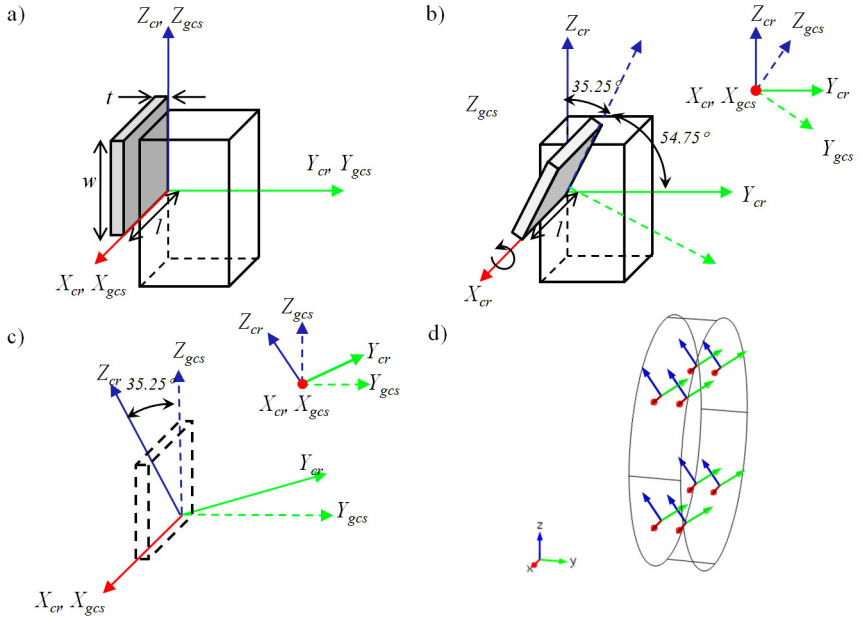


Figure 2-21: Defining an AT cut crystal plate within COMSOL, with normal in the global Y direction. Within the 1978 IEEE standard the AT cut is defined as (YXl) -35.25°. Start with the plate normal or thickness in the Y_{cr} direction (a) and rotate the plate –35.25° about the l-axis (b). The global coordinate system rotates with the plate. Finally rotate the entire system so that the global coordinate system is oriented as it appears in COMSOL (c). The local coordinate system should be defined with the Euler angles (ZXZ: 0, 35.25°, 0). (d) shows a coordinate system for this system in COMSOL.

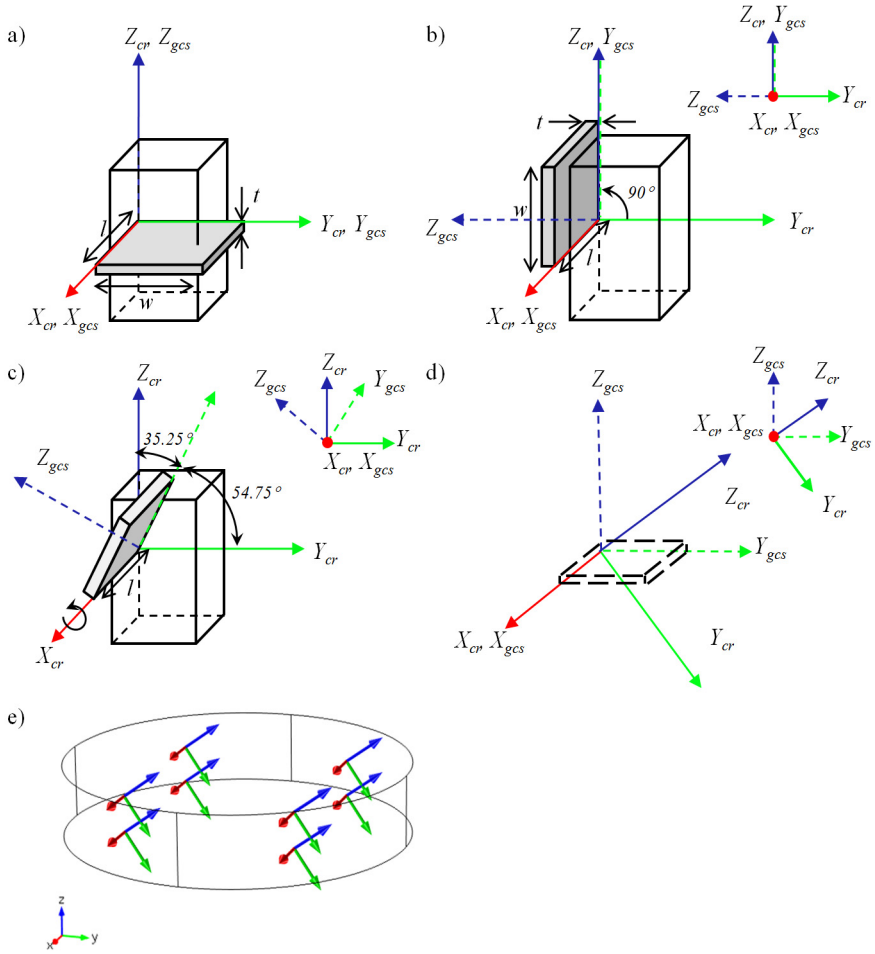


Figure 2-22: Defining an AT cut crystal plate within COMSOL, with normal in the global Z direction. Within the 1978 IEEE standard the AT cut is defined as (YXl) – 35.25°. Begin with the plate normal in the Z_{cr} direction, so the crystal and global systems are coincident. Rotate the plate so that its thickness points in the Y_{cr} direction (the starting point for the IEEE definition), the global system rotates with the plate (b). Rotate the plate –35.25° about the l-axis (d). Finally rotate the entire system so that the global coordinate system is oriented as it appears in COMSOL (d). The local coordinate system should be defined with the Euler angles (ZXZ: 0, -54.75°, 0). (e) shows a coordinate system for this system in COMSOL.

Piezoelectric Losses

Losses in piezoelectric materials can be generated both mechanically and electrically.

In the frequency domain, these can be represented by introducing complex material properties in the elasticity and permittivity matrices. Taking the mechanical case as an example, this introduces a phase lag between the stress and the strain, which corresponds to a Hysteretic Loss. These losses can be added to the Piezoelectric Material by three subnodes: Mechanical Damping, Coupling Loss, and Dielectric Loss. The losses typically defined as loss factors (see below).

The hysteretic electrical losses are usually used to represent high-frequency electrical losses that occur as a result of friction impeding the rotation of the microscopic dipoles that produce the material permittivity.

Low-frequency losses, corresponding to a finite material conductivity, can be added to the model through an Electric Conductivity (Time Harmonic) subnode. This feature operates only in the frequency domain.

In the time domain, the losses can be added by:

- Using the **Rayleigh Damping** option in the **Mechanical Damping** and **Coupling Loss** subnodes to the **Piezoelectric Material**,.

	Rayleigh Damping

- Using either the **Dispersion**, or **Complex permittivity**, or **Maximum Loss Tangent** option in the **Dielectric Loss** subnode to the **Piezoelectric Material**.

These types of damping are also available in the frequency domain.

	Rayleigh Damping

HYSTERETIC LOSS

In the frequency domain, the dissipative behavior of the material can be modeled using complex-valued material properties, irrespective of the loss mechanism. Such *hysteretic losses* can be applied to model both electrical and mechanical losses. For more information about hysteretic losses, see Ref. 1 to Ref. 4.

For the case of piezoelectric materials, this means that the constitutive equations are written as follows.

For the stress–charge formulation

$$\mathbf{T} = \tilde{c}_{\mathrm{E}}\mathbf{S} - \tilde{e}^{T}\mathbf{E}$$
$$\mathbf{D} = \tilde{e}\mathbf{T} + \varepsilon_0\tilde{\varepsilon}_{\mathrm{rS}}\mathbf{E}$$

where $\mathbf{T}$ is the stress tensor, and $\mathbf{S}$ is the strain tensor.

For the strain–charge formulation,

$$\mathbf{S} = \tilde{s}_{\mathrm{E}}\mathbf{T} + \tilde{d}^{T}\mathbf{E}$$
$$\mathbf{D} = \tilde{d}\mathbf{S} + \varepsilon_0\tilde{\varepsilon}_{\mathrm{rT}}\mathbf{E}$$

All constitutive matrices in the above equation can be complex-valued matrices, where the imaginary parts define the dissipative function of the material.

Both the real and complex parts of the material data must be defined so as to respect the symmetry properties of the material being modeled and with restrictions imposed by the laws of physics.

A key requirement is that the dissipation density is positive; that is, there is no power gain from the passive material. This requirement sets rules for the relative magnitudes for all material parameters. This is important when defining the coupling losses.

In COMSOL, you can enter the complex-valued data directly or by means of loss factors. When loss factors are used, the complex data $\tilde{X}$ is represented as pairs of a real-valued parameter

$$X = \mathrm{real}(\tilde{X})$$

and a loss factor

$$\eta_X = \mathrm{imag}(\tilde{X})/\mathrm{real}(\tilde{X})$$

the ratio of the imaginary and real part, and the complex data is then:

$$\tilde{X} = X(1 \pm j\eta_X)$$

where the sign depends on the material property used. The loss factors are specific to the material property, and thus they are named according to the property they refer to, for example, η_{cE}. For a structural material without coupling, simply use η_s, the structural loss factor.

The loss factors are defined so that a positive loss factor value usually corresponds to a positive loss. The complex-valued data is then based on sign rules.

By default, there is no damping until at least one of the damping and losses related subnodes is added.

For the Piezoelectric Material node, the following equations apply via the corresponding three subnodes:

Mechanical Damping

$$\tilde{c}_E^{m,n} = c_E^{m,n}(1 + j\eta_{cE}^{m,n})$$
$$\tilde{s}_E^{m,n} = s_E^{m,n}(1 - j\eta_{sE}^{m,n})$$

where m and n refer to components of each matrix.

Coupling Loss

$$\tilde{e}^{m,n} = e^{m,n}(1 + j\eta_e^{m,n})$$
$$\tilde{d}^{m,n} = d^{m,n}(1 + j\eta_d^{m,n})$$

Dielectric Loss

$$\tilde{\varepsilon}_{rS}^{m,n} = \varepsilon_{rS}^{m,n}(1 - j\eta_{\varepsilon S}^{m,n})$$
$$\tilde{\varepsilon}_{rT}^{m,n} = \varepsilon_{rT}^{m,n}(1 - j\eta_{\varepsilon T}^{m,n})$$

Note that the multiplication is applied component-wise.

The loss factors can also be entered as scalar isotropic factors independently of the material and the other coefficients.

A good check on the chosen values is to compute a number of eigenfrequencies, possibly using some different sets of boundary conditions. All computed eigenfrequencies must have a positive imaginary part in order to represent a damped motion.

In practice, it is often difficult to find complex-valued data for each of the matrix elements in the literature. Measuring the losses independently is a challenging task.

Two important particular cases are available in COMSOL, for which the dielectric loss data can be entered based on experimentally measurable quantities:

- Complex Permittivity
- Maximum Loss Tangent

DIELECTRIC DISPERSION

The **Dielectric Loss** subnode can be set to use the **Dispersion** option. In such case, the following equations need to be solved in the time domain:

$$\nabla \cdot \left(\frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}_\mathrm{p} \right) = 0 \tag{2-8}$$

$$\mathbf{J}_\mathrm{p} + \tau_\mathrm{d} \frac{\partial \mathbf{J}_\mathrm{p}}{\partial t} = \varepsilon_0 \Delta \varepsilon_{\mathrm{rS}} \frac{\partial \mathbf{E}}{\partial t} \tag{2-9}$$

where you can specify two material parameters: the relaxation time τ_d and the relative permittivity increment $\Delta\varepsilon_{\mathrm{rS}}$. The latter can be either a matrix or a scalar quantity. This model is a one-term version of the more general Debye dispersion model, Ref. 13.

The constitutive relation is assumed as

$$\mathbf{D} = e\mathrm{S} + \varepsilon_0 \varepsilon_\infty \mathbf{E} \tag{2-10}$$

where S is the strain tensor, and ε_∞ is the relative permittivity in the high-frequency limit (that is, for excitations with a characteristic time much shorter than the relaxation time τ_d).

The parent **Piezoelectric Material** node has an input for the relative permittivity, $\varepsilon_{\mathrm{rS}}$, which is used in stationary studies. You can choose how this input will be interpreted in the dispersion computations. The options are:

- High-frequency limit. In this case, $\varepsilon_\infty = \varepsilon_{\mathrm{rS}}$ is used.
- Low-frequency limit. In this case, $\varepsilon_\infty = \varepsilon_{\mathrm{rS}} - \Delta\varepsilon_{\mathrm{rS}}$ is used. Note that for consistency, ε_∞ must always be positive valued.

With the absence of free electric charges, Equation 2-8 and Equation 2-9 can be combined and integrated in time to yield the following equation:

$$\nabla \cdot \left(\mathbf{D} + \tau_d \frac{\partial \mathbf{D}}{\partial t} + \varepsilon_0 \Delta \varepsilon_{rS} \mathbf{E} \right) = 0$$

This is the equation form used in COMSOL Multiphysics for time-dependent analysis.

For the eigenfrequency and frequency domain analyses, the corresponding equation is:

$$\nabla \cdot [(1 + j\tau_d \omega) j\omega \mathbf{D} + \varepsilon_0 \Delta \varepsilon_{rS} j\omega \mathbf{E}] = 0$$

In most cases, $i\omega$ can be factored out, so that the following equation is solved:

$$\nabla \cdot (\mathbf{D} + j\tau_d \omega \mathbf{D} + \varepsilon_0 \Delta \varepsilon_{rS} \mathbf{E}) = 0$$

This equation, together with the constitutive relation Equation 2-10 gives

$$\nabla \cdot [e\mathrm{S} + \varepsilon_0 (\varepsilon' - j\varepsilon'') \mathbf{E}] = 0 \qquad (2\text{-}11)$$

where

$$\varepsilon' = \varepsilon_\infty + \frac{\Delta \varepsilon_{rS}}{1 + (\tau_d \omega)^2}$$

and

$$\varepsilon'' = \frac{\tau_d \omega \varepsilon_0 \Delta \varepsilon_{rS}}{1 + (\tau_d \omega)^2}$$

Equation 2-11 shows how the dispersion parameters contribute to the polarization and losses. Thus, the effective relative permittivity decreases with the excitation frequency from the low-frequency limit $\varepsilon_\infty + \Delta \varepsilon_{rS}$ down to the high-frequency limit ε_∞. The damping effect vanishes for both large and small frequencies, and it reaches the maximum for $\omega = 1/\tau_d$.

If your license includes either the AC/DC Module or MEMS Module, more options for modeling dielectric dispersion can be found in Dispersion section in the *AC/DC Module User's Guide*.

The following two sections present two cases, for which the dielectric dispersion data can be related to other experimentally measurable quantities. Both cases, can be used

in Eigenfrequency, Frequency Domain, and Time Dependent study. The software will apply the dispersion model equations, for which the effective relaxation time and relative permittivity increment are computed automatically based on the node input parameters.

Complex Permittivity

In this case, the complex relative permittivity is known at a certain reference frequency

$$\varepsilon'(f_{\mathrm{ref}}) - j\varepsilon''(f_{\mathrm{ref}})$$

The relaxation time and relative permittivity contribution cane be computed as

$$\tau_{\mathrm{d}} = \omega_{\mathrm{ref}}^{-1}\varepsilon''(\varepsilon' - \varepsilon_{\infty})^{-1}$$

$$\Delta\varepsilon_{\mathrm{rS}} = \varepsilon' - \varepsilon_{\infty} + (\varepsilon'')^{2}(\varepsilon' - \varepsilon_{\infty})^{-1}$$

where $\omega_{\mathrm{ref}} = 2\pi f_{\mathrm{ref}}$.

If the relative permittivity $\varepsilon_{\mathrm{rS}}$ (input on the **Piezoelectric Material** parent node) is selected to represent the low-frequency limit, one has

$$\varepsilon_{\infty} = \varepsilon' - (\varepsilon'')^{2}(\varepsilon_{\mathrm{rS}} - \varepsilon')^{-1}$$

If $\varepsilon_{\mathrm{rS}}$ is selected to represent the high-frequency limit, one can simply use $\varepsilon_{\infty} = \varepsilon_{\mathrm{rS}}$ instead.

Maximum Loss Tangent

The loss tangent is defined as a function of the frequency

$$\eta(f) = \tan\delta = \varepsilon''/\varepsilon'$$

so that the complex relative permittivity can be written as

$$\varepsilon_r(f) = \varepsilon'(\mathrm{I} - j\eta)$$

For many materials, the loss tangent reaches a maximum at certain frequency f_{ref} within the frequency range of interested

$$\eta_{\max} = \max\eta(f) = \eta(f_{\mathrm{ref}})$$

The relaxation time cane be computed as

$$\tau_{\mathrm{d}} = \omega_{\mathrm{ref}}^{-1}(\eta_{\max} + (\eta_{\max}^{2} + \mathrm{I})^{1/2})$$

where $\omega_{ref} = 2\pi f_{ref}$.

If the relative permittivity ε_{rS} (input on the **Piezoelectric Material** parent node) is selected to represent the low-frequency limit, the relative permittivity contribution is computes as

$$\Delta\varepsilon_{rS} = \varepsilon_{rS} - \varepsilon_{\infty}$$

where $\varepsilon_{\infty} = (\tau_d \omega_{ref})^{-2} \varepsilon_{rS}$.

If the relative permittivity ε_{rS} is selected to represent the high-frequency limit, it is computed as

$$\Delta\varepsilon_{rS} = ((\tau_d \omega_{ref})^2 - 1)\varepsilon_{\infty}$$

where $\varepsilon_{\infty} = \varepsilon_{rS}$.

ELECTRIC CONDUCTIVITY (TIME HARMONIC)

For frequency domain and eigenfrequency analyses, the effect of electric conductivity of the piezoelectric material (see Ref. 2, Ref. 5, and Ref. 6) can be included. Thus, in addition to the displacement current, the conduction electric current term is used:

$$\nabla \cdot (j\omega \mathbf{D} + \mathbf{J}_e) = 0$$

$$\mathbf{J}_e = \sigma_e \mathbf{E}$$

where σ_e is the material electric conductivity, and $\mathbf{E}$ is the electric field. The above form of the equation is used for the eigenfrequency analysis in COMSOL Multiphysics.

	When the conduction loss is applied, the default eigenvalue analysis will in most cases return a number of pure imaginary eigenfrequencies. To avoid this, you can configure the solver to search for eigenvalues with real part larger than zero. Do not use any dielectric loss factor damping together with the conduction loss in an eigenfrequency analysis.

	In the *COMSOL Multiphysics Reference Manual:* • Selecting a Stationary, Time-Dependent, or Eigenvalue Solver • Eigenvalue Solver

For the frequency domain analysis, the angular frequency is just a parameter, and the equation can be transformed into

$$\nabla \cdot (\mathbf{D} - j\omega^{-1}\sigma_e\mathbf{E}) = 0$$

which allows you to use both a dielectric loss factor and electric conductivity in a frequency response study. In such case, ensure that the loss factor refers to the alternating current loss tangent, which dominates at high frequencies, where the effect of ohmic conductivity vanishes (Ref. 7).

Conduction loss can be combined with Dielectric Dispersion for both eigenfrequency and frequency domain analyses. The following equation forms are used, respectively, in the frequency domain:

$$\nabla \cdot [\mathbf{D} + j\tau_d\omega\mathbf{D} + \varepsilon_0\Delta\varepsilon_{rS}\mathbf{E} + (\tau_d - j\omega^{-1})\mathbf{J}_e] = 0$$

and in eigenfrequency analyses:

$$\nabla \cdot [(1 + j\tau_d\omega)j\omega\mathbf{D} + \varepsilon_0\Delta\varepsilon_{rS}j\omega\mathbf{E} + (1 + j\tau_d\omega)\mathbf{J}_e] = 0$$

References for Piezoelectric Damping

1. R. Holland and E.P. EerNisse, *Design of Resonant Piezoelectric Devices*, Research Monograph No. 56, The M.I.T. Press, 1969.

2. T. Ikeda, *Fundamentals of Piezoelectricity*, Oxford University Press, 1990.

3. A.V. Mezheritsky, "Elastic, Dielectric, and Piezoelectric Losses in Piezoceramics: How it Works all Together," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 6, 2004.

4. K. Uchino and S. Hirose, "Loss Mechanisms in Piezoelectrics: How to Measure Different Losses Separately," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 1, pp. 307–321, 2001.

5. P.C.Y. Lee, N.H. Liu, and A. Ballato, "Thickness Vibrations of a Piezoelectric Plate With Dissipation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 1, 2004.

6. P.C.Y. Lee and N.H. Liu, "Plane Harmonic Waves in an Infinite Piezoelectric Plate With Dissipation," *Frequency Control Symposium and PDA Exhibition*, pp. 162–169, IEEE International, 2002.

7. C.A. Balanis, "Electrical Properties of Matter," *Advanced Engineering Electromagnetics*, John Wiley & Sons, chapter 2, 1989.

8. "Standards on Piezoelectric Crystals, 1949," *Proceedings of the I. R. E.*,vol. 37, no.12, pp. 1378–1395, 1949.

9. IEEE Standard on Piezoelectricity, ANSI/IEEE Standard 176-1987, 1987.

10. IEEE Standard on Piezoelectricity, ANSI/IEEE Standard 176-1978, 1978.

11. B.A. Auld, *Acoustic Fields and Waves in Solids*, Krieger Publishing, 1990.

12. R. Bechmann "Elastic and Piezoelectric Constants of Alpha-Quartz," *Physical Review B*, vol. 110 no. 5, pp. 1060–1061, 1958.

13. N.S. Stoykov, T.A. Kuiken, M.M. Lowery, and A. Taflove, "Finite-element time-domain algorithms for modeling linear Debye and Lorentz dielectric dispersions at low frequencies," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 9, pp. 1100–1107, 2003.

Piezoelectric Coupling

The piezoelectric effect is an interaction between the mechanical and electrical physics, where a stress applied on a piezoelectric material generates a voltage (direct effect) or a voltage applied on it generates the deformation of the material (inverse effect). In COMSOL Multiphysics, the Piezoelectricity interface is constituted of one **Solid Mechanics** and one **Electrostatics** interface, which are coupled together by a **Piezoelectricity** multiphysics feature. Hence a piezoelectric problem contains solid and electrostatic domains, with at least one domain shared by the two physics interfaces and with the piezoelectric coupling defined on it.

Create the Piezoelectricity Interface and Define Domains

A piezoelectric problem can be set up in different ways:

- By selecting **Piezoelectricity** from the **Model Wizard**,
- By choosing **Piezoelectricity** from the **Add Physics** menu when working in an existing model, or
- By adding the corresponding features to create the coupling manually.

In the first two cases, by default all the domains in the model are assumed to be piezoelectric materials.

When setting up the problem manually (that is, by adding single physics interfaces one at a time) both **Solid Mechanics** and **Electrostatics** interfaces should be added. Then, you have to specify which domains are in each physics, and which domains are to be modeled as piezoelectric materials.

1. On the **Solid Mechanics** interface **Settings** window, locate the **Domain Selection** section. Select the domains which undergo structural deformation, including the piezoelectric material domains.
2. Go to the **Solid Mechanics** > **Piezoelectric Material** node (if it is not yet available, add it). Select the domains where the piezoelectric effect applies. Domains which are not piezoelectric can be modeled using other available material models.
3. Go to the **Electrostatics** node and under **Domain Selection** select the domains the electrostatics equations must be solved. These domains include all the piezoelectric domains, and any other insulating domains.

Since metals usually have a conductivity several orders of magnitude higher than other materials, it is normal not to solve the electrostatics equations in their corresponding domains, but instead the surfaces of the material are represented as isopotentials with an appropriate boundary condition (usually a terminal or floating potential).

4. Go to the **Electrostatics** > **Charge Conservation, Piezoelectric** node (if it is not yet available, add it). Select the domains where the piezoelectric effect has to be modeled.
5. A **Multiphysics** > **Piezoelectricity** node is already present if the coupling was added using either the **Model Wizard** or **Add Physics** window. If the model is set up manually (that is, single physics interfaces are added), right-click the **Multiphysics Couplings** node to add a **Piezoelectricity** coupling.
6. Confirm that all the domains where the piezoelectric material is present are selected.

Only domains that have both **Charge Conservation, Piezoelectric** selected in the **Electrostatics** interface and **Piezoelectric Material** selected in the **Solid Mechanics** interface are selected. The selection of this feature cannot be edited. If several **Solid Mechanics** or **Electrostatics** interfaces are present, select the correct ones.

7 Confirm that remaining domains are well assigned:

- Solid and electrostatic domains: In these domains, the electrostatics and structural problems are solved independently without any piezoelectric coupling. This is the case for insulators.
- Solid–only domains: In these domains, only mechanical phenomena are modeled, and the electrostatics phenomena are neglected. Metals are typically modeled in this way, because their conductivity is so high that their surfaces can be treated as isopotential surfaces within the electrostatics problem. In some cases, insulators are modeled using these settings, for example, when there is no potential applied across the domain, and correspondingly solving the electrostatics equations would produce a constant potential and waste computing resources.
- Electrostatics-only domains: Those domains are nonsolid. This is typically the case of air, in which the electrostatics is solved but not mechanics.
- Nonsolid and nonelectrostatic domains: In those domains, another physics is solved. A typical example is an acoustic domain in the case of piezoacoustic modeling.

Complete Settings of Piezoelectric Materials

Go to the **Solid Mechanics** > **Piezoelectric Material** node. In the **Settings** window, complete these settings:

- **Coordinate System Selection** section: The material is poled in the **x3** direction of the coordinate system **(x1, x2, x3)** specified in this section. By default, it is set to the global coordinate system. If the piezoelectric material is poled along another direction, you need to define a coordinate system so that its third direction is aligned with the poling direction. Then, assign it as the coordinate system which orients the material in the Coordinate System Selection section.

If a given piezoelectric material is present with several orientations (such as stacked piezoelectric disks) you need to define several **Piezoelectric Material** nodes and to assign a different coordinate system for each of them.

- **Piezoelectric Material Properties**: Select whether the constitutive relation of piezoelectric material is in stress–charge or in strain–charge form. This choice defines the type of material properties that will be used.
- **Geometric Nonlinearity**: select the checkbox to force strains to be linear.

- **Energy Dissipation**: Select the checkbox to enable the calculation of the dissipated energy.

Add Damping and Loss

In the physics toolbar you can add attributes to the Piezoelectric Material node, especially the following damping and loss contributions:

- **Mechanical Damping**: Specify the domains of application, then choose if you want to define a loss factor for c_E, a loss factor for s_E (in strain–charge form), an isotropic loss factor, or a Rayleigh damping.
- **Dielectric Loss**: Specify the domains of application, then choose if you want to define a loss factor for ε_{rS}, a loss factor for ε_{rT} (in strain–charge form), or dispersion.
- **Coupling Loss**: Specify the domains of application, then choose if you want to define a loss factor for e, a loss factor for d (in strain–charge form), or Rayleigh damping.
- **Conduction Loss (Time-Harmonic)**: Specify the domains of application, then choose how you want to define the electric conductivity.

Define Material Properties

Define material properties for the piezoelectric materials. The material library contains several common piezoelectric materials under the piezoelectric section. If you want to define your own piezoelectric material, you need to specify its properties by hand. The required properties depend on whether the constitutive relations are in stress–charge or strain–charge form, and which damping and loss attributes are created. Defining all piezoelectric settings before materials preselects the required properties and makes the completion easier:

MECHANICAL PROPERTIES

- Density rho (SI unit: kg/m^3)
- Elasticity matrix c_E (SI unit: Pa) when **Stress–charge form** is selected.
- Compliance Matrix s_E (SI unit: $1/Pa$) when **Strain–charge form** is selected.

ELECTROSTATIC PROPERTIES

- Relative Permittivity ε_{rS} (dimensionless) when **Stress–charge form** is selected.
- Relative Permittivity ε_{rT} (dimensionless) when **Strain–charge form** is selected.

COUPLING PROPERTIES

- Coupling matrix e_{ES} (SI unit: C/m^2) when **Stress–charge form** is selected.
- Coupling matrix d_{ET} (SI unit: C/N) when **Strain–charge form** is selected.

DAMPING AND LOSS PROPERTIES

- Either loss factor η_{cE} for elasticity matrix c_E, or the loss factor η_{sE} for compliance matrix s_E: required when **Mechanical Damping** is present. Both are dimensionless, the latter is valid only when **Strain–charge form** is selected.
- Either loss factor $\eta_{\varepsilon S}$ for electric permittivity matrix ε_{rS}, or loss factor $\eta_{\varepsilon T}$ for electric permittivity matrix ε_{rT}: required when **Dielectric Loss** is present. Both are dimensionless, the latter is valid only when **Strain–charge form** is selected.
- Either loss factor η_e for coupling matrix e, or loss factor η_d for coupling matrix d: required when **Coupling Loss** is present. Both are dimensionless, the latter is valid only when **Strain–charge form** is selected.
- Electric conductivity σ_e (SI unit: S/m): required when **Conduction Loss** is present.

Working with Materials and Piezoelectric Materials Library in the *COMSOL Multiphysics Reference Manual.*

Coupling Piezoelectricity with Acoustics

Using piezoelectric materials for an acoustic application is common, such as in sonars, microphones, sensors, and so forth. This is why coupling piezoelectric devices with acoustic domains is of particular interest for these applications.

Compared to a single piezoelectric model, you need to add a pressure acoustics interface, for example, **Pressure Acoustics, Frequency Domain** or **Pressure Acoustics, Transient** (depending on which study type you want to use) plus an **Acoustic–Structure Boundary** coupling under the **Multiphysics Couplings** node. You can also directly create the nodes that are needed for coupling by adding an **Acoustic-Piezoelectric Interaction** interface from the **Model Wizard** or **Add Physics** windows. If solid and acoustic domains are correctly defined, then the right coupling boundaries are automatically selected. Then specify domains of application for each physics.

- Select solid domains and **Piezoelectric Material** domains in **Solid Mechanics**.
- Select electrostatic domains and **Charge Conservation, Piezoelectric** domains in **Electrostatics**.

- Select acoustic domains in a **Pressure Acoustics** node.
- Under the **Multiphysics** branch, confirm that selections for **Piezoelectricity** and **Acoustic–Structure Boundary** are the right ones. If several **Pressure Acoustics**, **Solid Mechanics**, or **Electrostatics** interfaces are present, select the right ones that need to be coupled in the multiphysics interfaces.
- Continue the modeling process by entering the settings for each physics interface and feature and define materials.

Modeling Electrostrictive and Ferroelectroelastic Materials

Electrostriction Coupling

Electrostriction is an interaction between the mechanical and electric physics, where an electric field applied on electrostrictive material generates the deformation of the material (direct effect), and a stress applied on it changes the material polarization (inverse effect). In contrast to linear piezoelectricity, the electrostrictive strain induced in the polarized material is proportional to the square of the polarization. Thus, the strain will remain the same if the direction of the applied electric field is changed to the opposite.

Ferroelectricity

The ferroelectroelasticity and ferroelectricity phenomena are related to phase transitions in materials. In its ferroelectric phase, the material exhibits spontaneous polarization, so that it is constituted of domains with nonzero polarization even at zero applied field. This is similar to permanent magnetism in ferromagnetics, which explains the name used for such materials. Electrostriction in ferroelectroelastic materials can be related to the domain rotation. Thus, the applied electric field can both rearrange the domains resulting into the net polarization and rotate the domains mechanically. Thus, the material extends in the direction of the electric field and contracts in the direction perpendicular to the field. The domain rotation can be affected by an applied mechanical stress, which also results into the effective polarization. At very large electric fields, the electrostrictive effect saturates, as all ferroelectric domains in the material are aligned along the direction of the applied field. Domain wall interactions can also lead to a significant hysteresis in the polarization and strain.

Many piezoelectric materials exhibit such nonlinear ferroelectroelastic behavior at large applied electric fields.

Multiphysics Interfaces

In COMSOL Multiphysics, the electrostrictive effect can be modeled using two multiphysics interfaces: **Electrostriction** and **Ferroelectroelasticity**. Both interfaces are constituted of one **Solid Mechanics** and one **Electrostatics** interface, which are coupled

together via an **Electrostriction** multiphysics feature. However, in case of **Ferroelectroelasticity**, the corresponding **Electrostatics** interface will contain a dedicated feature **Charge Conservation, Ferroelectric**.

The **Electrostriction** interface electrostatically can be used for modeling electrostatically actuated structures in regimes when the electric polarization can be assumed to vary linear with the applied electric field.

The **Ferroelectroelasticity** can be used for modeling nonlinear electromechanical interaction in ferroelectric and piezoelectric materials at high applied electric fields. Electric polarization in such materials depends nonlinearly on the applied electric field including possible hysteresis and saturation effects.

Ferroelectricity in the *AC/DC Module User's Guide*.

Electrostriction and Ferroelectroelasticity in the *Structural Mechanics Module User's Guide*.

Modeling Magnetostrictive Materials

Magnetostriction Coupling

Magnetostriction is an interaction between the mechanical and magnetic physics, where a magnetic field applied on magnetostrictive material generates the deformation of the material (direct effect or Joule effect), while a mechanical stress applied on it changes the material magnetization (inverse effect or Villari effect).

The magnetostrictive strain has a nonlinear dependence on the magnetic field and the mechanical stress in the material. However, the effect can be modeled using linear coupled constitutive equations if the response of the material consists of small deviations around an operating point (bias point). This type of coupling is referred to as Piezomagnetism.

In COMSOL Multiphysics, there are two multiphysics interfaces for modeling either linear or nonlinear magnetostriction, respectively:

- The Piezomagnetism Interface
- The Nonlinear Magnetostriction Interface

Both multiphysics interfaces, the Magnetostriction interfaces are constituted of one **Solid Mechanics** and one **Magnetic Fields** interface, which are coupled together via either Piezomagnetism or Nonlinear Magnetostriction multiphysics coupling feature. Hence, a magnetostriction problem contains solid and magnetic domains, with at least one domain shared by the two physics interfaces and with the magnetostriction coupling defined on it.

Linear vs. Nonlinear Magnetostriction

For the piezomagnetic model, it is possible to express the relation between the stress $\mathbf{S}$, strain ε, magnetic field $\mathbf{H}$, and magnetic flux density $\mathbf{B}$ in either a *stress–magnetization form* or *strain–magnetization form*. In COMSOL Multiphysics, both constitutive forms can be used; simply select one and the software will make any necessary transformations. You find all the required material data inputs within the **Piezomagnetic Material** selected in the **Solid Mechanics** interface.

The nonlinear model of magnetostrictive strain can be used for the whole range from full demagnetization to saturation magnetization. In the case of nonlinear magnetostriction, the magnetization model can be selected. The following options are

available: Langevin function, Hyperbolic tangent, User defined, and Linear. The last option will make it possible to find an explicit expression for the magnetization. Note however that such model does not have a proper saturation behavior, and thus it should be used only in the operating range far from saturation. For all other choices, the magnetization vector components will be treated as extra dependent variables.

For more details, see the corresponding theory section in Magnetostriction and Piezomagnetism.

Create the Magnetostriction Interface and Define Domains

A magnetostriction problem can be set up in different ways:

- By selecting either **Piezomagnetism** or **Nonlinear Magnetostriction** from the **Model Wizard**,
- By choosing either **Piezomagnetism** or **Nonlinear Magnetostriction** from the **Add Physics** menu when working in an existing model, or
- By adding the corresponding coupling features to create the coupling manually.

In the first two cases, by default all the domains in the model are assumed to be magnetostrictive materials.

PIEZOMAGNETISM

When setting up the problem manually (that is, by adding single physics interfaces one at a time) both **Solid Mechanics** and **Magnetic Fields** interfaces should be added. Then, you have to specify which domains are in each physics, and which domains are to be modeled as piezomagnetic materials.

1. On the **Solid Mechanics** interface **Settings** window, locate the **Domain Selection** section. Select the domains which undergo structural deformation, including the piezomagnetic material domains.
2. Go to the **Solid Mechanics** > **Piezomagnetic Material** node. If it is not yet available, add such node. Select the domains where the piezomagnetic effect applies. You find all the necessary material data inputs within the **Piezomagnetic Material**. This includes the elasticity, magnetic permeability and coupling matrices. Nonpiezomagnetic domains can be modeled using any other available structural material model.
3. Go to the **Magnetic Fields** node and under **Domain Selection** select the domains the magnetics equations must be solved. These domains include all the piezomagnetic domains, and any other magnetic domains.

4 Go to the **Magnetic Fields** > **Ampère's Law, Piezomagnetic** node. If it is not yet available, add such node. Select the domains where the piezomagnetic effect needs to be modeled.

5 A **Multiphysics** > **Piezomagnetism** node is already present if the coupling was added using either the **Model Wizard** or **Add Physics** window. If the model is set up manually (that is, single physics interfaces are added), right-click the **Multiphysics Couplings** node to add a **Piezomagnetism** coupling. If several **Solid Mechanics** or **Magnetic Fields** interfaces are present, select the correct ones. By default, the model will be solved as fully coupled. Alternatively, on the **Piezomagnetism** coupling node, you can select to include only either Joule effect or Villari effect. Confirm that all the domains where the piezomagnetic material is present are selected.

Only domains that have both **Ampère's Law, Piezomagnetic** selected in the **Magnetic Fields** interface and **Piezomagnetic Material** selected in the **Solid Mechanics** interface are applicable in the selection for **Piezomagnetism** coupling. The selection of this multiphysics coupling node cannot be edited directly.

6 Confirm that remaining domains are well assigned:

- Solid and magnetic domains: In these domains, the magnetics and structural problems are solved independently without any piezomagnetic coupling. This represents deformable magnetic materials for which such coupling is negligible. Exclude such domains from the selections for all **Ampère's Law, Piezomagnetic**, and **Piezomagnetic Material** nodes under the corresponding interfaces.
- Solid–only domains: In these domains, only mechanical phenomena are modeled, and the magnetics phenomena are neglected. This represents deformable solid nonmagnetic materials. Exclude such domains from the selection in the **Magnetic Fields** interface.
- Magnetics-only domains: Those domains are nonsolid. This is typically the case of air, in which the magnetics is solved but not mechanics. Exclude such domains from the selection in the **Solid Mechanics** interface.
- Nonsolid and nonmagnetic domains: In those domains another physics is solved. Typical examples are acoustic domains and fluid flow domains, which might be

present in certain multiphysics application. Exclude such domains from the selections on both **Magnetic Fields** and **Solid Mechanics** interface nodes.

- Piezomagnetic Material
- Ampère's Law, Piezomagnetic

NONLINEAR MAGNETOSTRICTION

When setting up the problem manually (that is, by adding single physics interfaces one at a time) both **Solid Mechanics** and **Magnetic Fields** interfaces should be added. Then, you have to specify which domains are in each physics, and which domains are to be modeled as magnetostrictive materials.

Instead of **Magnetic Fields** interface, you can also use **Rotating Machinery, Magnetic** interface.

1. On the **Solid Mechanics** interface **Settings** window, locate the **Domain Selection** section. Select the domains which undergo structural deformation, including the magnetostrictive material domains. The following structural material nodes are supported to represent the magnetostrictive domains: **Linear Elastic Material** and **Hyperelastic Material** (available with the Nonlinear Structural Materials Module).
2. Go to the **Magnetic Fields** node and under **Domain Selection** select the domains the magnetics equations must be solved. These domains include all the magnetostrictive domains, and any other magnetic domains.
3. Go to the **Magnetic Fields** > **Ampère's Law, Nonlinear Magnetostrictive** node. If it is not yet available, add such node. Select the domains where the magnetostriction effect needs to be modeled. You can specify the magnetization model and enter the related material data on the node.
4. A **Multiphysics** > **Nonlinear Magnetostriction** node is already present if the coupling was added using either the **Model Wizard** or **Add Physics** window. If the model is set up manually (that is, single physics interfaces are added), right-click the **Multiphysics Couplings** node to add a **Nonlinear Magnetostriction** coupling. If several **Solid Mechanics** or **Magnetic Fields** interfaces are present, select the correct ones. On the coupling node, you can also select the magnetostriction model depending on the material symmetry and enter the corresponding coupling data. By default, the model will be solved as fully coupled. Alternatively, on the **Nonlinear Magnetostriction**

coupling node, you can select to include only either Joule effect or Villari effect. Select only the domains where the magnetostrictive effect needs to be modeled.

On the **Nonlinear Magnetostriction** coupling node, the applicable domains in the selection are only domains that have both **Ampère's Law, Nonlinear Magnetostrictive** selected in the **Magnetic Fields** interface and **Linear Elastic Model** (or **Hyperelastic Model**) selected in the **Solid Mechanics** interface.

5 Confirm that remaining domains are well assigned:

- Solid and magnetic domains: In these domains, the magnetics and structural problems are solved independently without any magnetostriction coupling. This represents deformable magnetic materials for which such coupling is negligible. Exclude such domains from the selection in the **Nonlinear Magnetostriction** coupling node.
- Solid–only domains: In these domains, only mechanical phenomena are modeled, and the magnetics phenomena are neglected. This represents deformable solid nonmagnetic materials. Exclude such domains from the selection in the **Magnetic Fields** interface.
- Magnetics-only domains: Those domains are nonsolid. This is typically the case of air, in which the magnetics is solved but not mechanics. Exclude such domains from the selection in the **Solid Mechanics** interface.
- Nonsolid and nonmagnetic domains: In those domains another physics is solved. Typical examples are acoustic domains and fluid flow domains, which might be present in certain multiphysics applications. Exclude such domains from the selections in both **Magnetic Fields** and **Solid Mechanics** interfaces.

- Ampère's Law, Nonlinear Magnetostrictive

Mechanical Damping and Losses

Damping and losses are important factors for determining the response in time-dependent and frequency domain studies, and sometimes also when computing eigenfrequencies. This section describes how to model damping and loss using different damping models. In this section:

In this section:

- About Damping
- Rayleigh Damping
- Loss Factor Damping
- Viscoelastic Damping
- Explicit Complex-Valued Damping
- Equivalence Between Loss Factor, Rayleigh, and Viscous Damping
- Piezoelectric Damping
- Adding Damping in the Modal Solver

Damping Sources

There are many sources of damping in a system. Some of them are:

- Dissipation in the material. This dissipation can be for example be proportional to the amplitude of the strain rate (viscous damping) or to the amplitude of the strain itself (hysteretic damping).
- Thermoelastic damping, which is a thermodynamic effect which is related not only to the state in a point, but also to the gradients of the temperature field.
- Damping caused by the surrounding medium, often air or water.
- Friction between joined parts.
- Components intended to supply damping like a dashpot in a car suspension.

It is often difficult to separate and quantify these effects, so damping modeling is one of the biggest challenges in structural dynamics.

About Damping

Phenomenological damping models are typically invoked to model the intrinsic frictional damping present in most materials (material damping). These models are easiest to understand in the context of a system with a single degree of freedom. The following equation of motion describes the dynamics of such a system with viscous damping:

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = f(t) \tag{2-12}$$

In this equation u is the displacement of the degree of freedom, m is its mass, c is the damping parameter, and k is the stiffness of the system. The time (t) dependent forcing term is $f(t)$. This equation is often written in the form:

$$\frac{d^2u}{dt^2} + 2\zeta\omega_0\frac{du}{dt} + \omega_0^2 u = \frac{f(t)}{m} \tag{2-13}$$

where $\zeta = c/(2m\omega_0)$ and $\omega_0{}^2 = k/m$. In this case ζ is the *damping ratio* ($\zeta = 1$ for critical damping) and ω_0 is the *undamped resonant frequency* of the system. In the literature it is more common to give values of ζ than c. The damping ratio ζ can also be readily related to many of the various measures of damping employed in different disciplines. These are summarized in Table 2-12.

TABLE 2-12: RELATIONSHIPS BETWEEN MEASURES OF DAMPING.

DAMPING PARAMETER	DEFINITION	RELATION TO DAMPING RATIO
Damping ratio	$\zeta = c/c_{\text{critical}}$	–
Logarithmic decrement	$\delta_d = \ln\left(\frac{u(t_0)}{u(t_0+\tau)}\right)$ where t_0 is a reference time and τ is the period of vibration for a decaying, unforced degree of freedom.	$\delta_d \approx 2\pi\zeta$ $(\zeta \ll 1)$

TABLE 2-12: RELATIONSHIPS BETWEEN MEASURES OF DAMPING.

DAMPING PARAMETER	DEFINITION	RELATION TO DAMPING RATIO
Quality factor	$Q = \omega/\Delta\omega$ where $\Delta\omega$ is the bandwidth of the amplitude resonance measured at $1/\sqrt{2}$ of its peak.	$Q \approx 1/(2\zeta)$ $(\zeta \ll 1)$
Loss factor	$\eta = \frac{1}{2\pi}\left(\frac{Q_h}{W_h}\right)$ where Q_h is the energy lost per cycle and W_h is the maximum potential energy stored in the cycle. The variables Q_h and W_h are available as `solid.Qh` and `solid.Wh`.	At the resonant frequency: $\eta \approx 2\zeta$ $(\zeta \ll 1)$

In the frequency domain, the time dependence of the force and the displacement can be represented by introducing a complex force term and assuming a similar time dependence for the displacement. The equations

$$f(t) = \mathrm{Re}\{Fe^{j\omega t}\} \text{ and } u(t) = \mathrm{Re}\{Ue^{j\omega t}\}$$

are written where ω is the angular frequency and the amplitude terms U and F can in general be complex (the arguments provide information on the relative phase of signals). Usually the real part is taken as implicit and is subsequently dropped. Equation 2-12 takes the following form in the frequency domain:

$$-\omega^2 mU + j\omega cU + kU = F \tag{2-14}$$

where the time dependence has canceled out on both sides. Alternatively, this equation can be written as

$$-\omega^2 U + 2j\omega\omega_0\zeta U + \omega_0^2 U = \frac{F}{m} \tag{2-15}$$

There are three basic damping models available in the structural mechanics interfaces for explicit modeling of material damping — Rayleigh damping, viscous damping, and loss factor models based on introducing complex quantities into the equation system. There are also other phenomena which contribute to the damping. Some material models, such as viscoelasticity and plasticity are inherently dissipative. It is also possible to model damping in spring conditions.

Rayleigh Damping

A common method of modeling damping is *Rayleigh damping*, where two damping coefficients are specified. This type of damping is not directly related to any physical process, but must be seen as a way to take the total damping of a structure into account.

Rayleigh damping introduces damping in a form based on Equation 2-12. This means that the method can be applied generally in either the time or frequency domain. The parameter c in Equation 2-12 is defined as a fraction of the mass and the stiffness using two parameters, α_{dM} and β_{dK}, such that

$$c = \alpha_{dM} m + \beta_{dK} k \tag{2-16}$$

Substituting this relationship into Equation 2-12 and rearranging into the form of Equation 2-13 gives:

$$\frac{d^2u}{dt^2} + (\alpha_{dM} + \beta_{dK}\omega_0^2)\frac{du}{dt} + \omega_0^2 u = \frac{f(t)}{m}$$

When there are many degrees of freedom m, k, and c become matrices and the technique can be generalized.

Rayleigh damping can therefore be identified as equivalent to the damping ratio at resonance of:

$$\zeta = \frac{1}{2}\left(\frac{\alpha_{dM}}{\omega_0} + \beta_{dK}\omega_0\right) \tag{2-17}$$

Note that Equation 2-17 holds separately for each vibrational mode in the system at its resonant frequency. In the frequency domain it is possible to use frequency dependent values of α_{dM} and β_{dK}. For example, setting $\alpha_{dM} = 0$ and $\beta_{dK} = 2\zeta/\omega_0$ produces an equivalent viscous damping model at the resonant frequency ω_0.

While Rayleigh damping is numerically convenient, the model does not agree with experimental results for the frequency dependence of material damping over an extended range of frequencies. This is because the material damping forces behave more like frictional forces (which are frequency independent) than viscous damping forces (which increase linearly with frequency as implied by Equation 2-14). In the frequency domain it is possible to introduce loss factor damping, which has the desired property of frequency independence.

A complication with the Rayleigh damping model is to obtain good values for the damping parameters α_{dM} and β_{dK}. A more physical damping measure is the damping ratio, the ratio between actual and critical damping, often expressed as a percentage of the critical damping. Commonly used values of the damping ratio can be found in the literature.

Using Equation 2-17, this relationship at two frequencies, f_1 and f_2, with different damping ratio, ζ_1 and ζ_2, results in an equation system that can be solved for α_{dM} and β_{dK}:

$$\begin{bmatrix} \frac{1}{4\pi f_1} & \pi f_1 \\ \frac{1}{4\pi f_2} & \pi f_2 \end{bmatrix} \begin{bmatrix} \alpha_{dM} \\ \beta_{dK} \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}$$

The damping ratios are then

$$\alpha_{dM} = 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{dK} = \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi (f_2^2 - f_1^2)}$$

Using the same damping ratio, $\zeta_1 = \zeta_2 = \zeta_0$, does not result in a constant damping factor inside the interval $f_1 < f < f_2$. It can be shown that the damping factor is lower inside the interval, as Figure 2-23 shows.

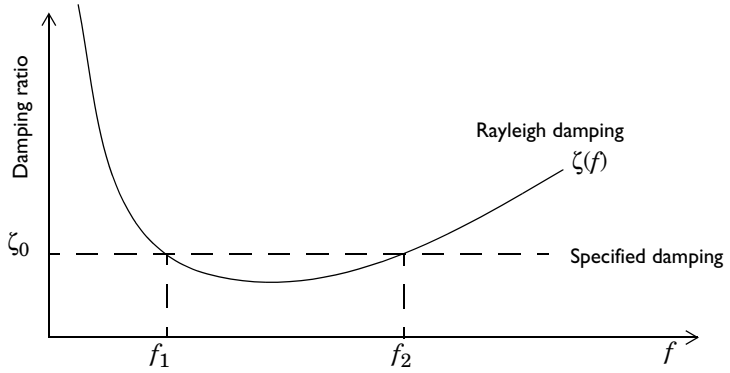


Figure 2-23: An example of Rayleigh damping.

Since the coefficients α_{dM} and β_{dK} should not be negative, the damping ratios are constrained by the respective frequencies as

$$\frac{f_1}{f_2} \leq \frac{\zeta_2}{\zeta_1} \leq \frac{f_2}{f_1}$$

For many applications it is sufficient to leave α_{dM} as zero and to define damping only using the β_{dK} coefficient. Then, according to Equation 2-17, a damping that increases linearly with frequency is obtained. If the damping ratio $\zeta(f_0)$ or loss factor $\eta(f_0)$ is known at a given frequency f_0, the appropriate value for β_{dK} is:

$$\beta_{dK} = \zeta/(\pi f_0) = \eta/(2\pi f_0)$$

In order to visualize the damping ratio as a function of frequency, click **Damping Ratio Preview** ( ).

Loss Factor Damping

Loss factor damping (sometimes referred to as material damping, structural damping, or hysteretic damping) can be applied in the frequency domain.

The loss factor is a measure of the inherent damping in a material when it is dynamically loaded. It is typically defined as the ratio of energy dissipated in unit volume per radian of oscillation to the maximum strain energy per reference volume.

In COMSOL Multiphysics the loss information appears as a multiplier to the elastic constitutive matrix D^{c}

$$D^{\mathrm{c}} = (1 + j\eta_{\mathrm{s}})D$$

For a nonlinear elastic material, this applies to the tangential stiffness.

The use of loss factor damping traditionally refers to a scalar-valued loss factor η_{s}. But there is no reason that η_{s} must be scalar. Because the loss factor is a value deduced from true complex-valued material data, it can be represented by a matrix of the same dimensions as the anisotropic stiffness matrix. Especially for orthotropic materials, there should be a set of loss factors of all normal and shear elasticity modulus components, and COMSOL allows all these options, so a more general expression is.

$$D^{\mathrm{c}}_{mn} = (1 + j\eta_{\mathrm{s},\,mn})D_{mn}$$

For hyperelastic materials the loss information appears as a stress contribution to the second Piola–Kirchhoff stress, S:

$$S_q = j\eta_s \frac{\partial W_s}{\partial E}$$

Loss factor damping is available for *frequency response* analysis and damped eigenfrequency analysis in all interfaces.

Viscous Damping

You can add an explicit viscous damping to several material models. Viscous damping can be used both in time domain and frequency domain. In the viscous damping model, and extra contribution, proportional to the strain rate, is added to the stress tensor, as described in Viscous Damping.

You can specify the viscous damping for volumetric strains and shear strains independently.

Equivalence Between Loss Factor, Rayleigh, and Viscous Damping

In frequency domain, it is possible to use Rayleigh damping in order to specify an equivalent viscous damping. Set the stiffness damping parameter β_{dK}, to the loss factor, η, divided by the excitation frequency:

$$\beta_{dK} = \frac{\eta}{2\pi f} = \frac{\eta}{\omega}$$

The mass damping factor, α_{dM}, should be set to zero.

If, on the other hand, you would want to use a viscous damping, corresponding to a certain Rayleigh stiffness damping, the conversion to bulk and shear viscosity can be made using the expressions

$$\eta_b = K\beta_{dK}$$
$$\eta_v = G\beta_{dK}$$

where K and G are the bulk and shear moduli, respectively. Equivalently, you can transform between loss factor damping and viscous damping,

$$\eta_b = \frac{K\eta}{\omega}$$
$$\eta_v = \frac{G\eta}{\omega}$$

Explicit Complex-Valued Damping

In frequency domain, it is possible to define damping by modeling the dissipative behavior of the material using complex-valued material properties. In COMSOL Multiphysics, you can enter the complex-valued data directly, using `i` or `sqrt(-1)` for the imaginary unit.

However, entering complex-valued data directly into the material parameters is incompatible with the built-in damping and viscoelasticity features.

Viscoelastic Damping

In some cases, damping is included implicitly in the material model. This is the case for a viscoelastic material, where damping operates on the shear components of stress and strain.

When viscoelasticity is modeled in the frequency domain, it will act as a loss factor damping. The complex modulus $G^*(\omega)$ is the frequency-domain representation of the stress relaxation function of viscoelastic material. It is defined as

$$G^* = G' + jG'' = (1 + j\eta_s)G'$$

where G' is the storage modulus, G'' is the loss modulus, and their ratio $\eta_s = G''/G'$ is the loss factor. The term G' defines the amount of stored energy for the applied strain, whereas G'' defines the amount of energy dissipated as heat; G', G'', and η_s can all be frequency dependent.

Piezoelectric Damping

Piezoelectric losses are more complex and include coupling and electrical losses in addition to the material terms. For damping in piezoelectric materials, see Piezoelectric Losses.

For piezoelectric materials, β_{dK} is only used as a multiplier of the structural contribution to the stiffness matrix when creating the damping matrix as given by

Equation 2-16. In the frequency domain studies, you can use the coupling and dielectric loss factors equal to $\beta_{dK}\omega$ to effectively achieve the Rayleigh damping involving the whole stiffness matrix.

Adding Damping in the Modal Solver

In COMSOL it is possible to solve a problem for a set of modes in the absence of damping, and then to use those solutions as a modal basis to solve a problem in the time (using a time domain modal study) or frequency domain (using a Frequency Domain, Modal study). In both of these cases it is possible to manually assign a damping ratio to the computed modes in the time or frequency domain study. To do that, right click on the study and choose **Show Default Solver**, then expand the solver sequence until the **Modal Solver** node is visible. In the Settings window for that node, add damping ratios for each of the modes.

For more details, see the section Modal Solver in the *COMSOL Multiphysics Reference Manual.*

Modeling Geometric Nonlinearity

This section discusses how to model problems where displacements or strains are of a size where the deformation of the structure has to be taken into account when formulating the equations. Examples of the type of problems where this feature is useful include:

- Thin structures, where the deflection is of the same order of magnitude as the thickness.
- Where the structure exhibits large rotations. A rigid body rotation of only a few degrees causes significant strains and stresses in a material where a linear strain representation is used.
- Where the strains are larger than a few percent.
- Contact problems.
- Where a prestress must be taken into account for computing the dynamic response of a structure.
- *Buckling* problems.
- Where a deformed mesh is used.
- Fluid–Structure interaction problems.

- Contact Modeling
- Fluid–Structure Interaction
- The more formal theory is described in Analysis of Deformation

Geometric Nonlinearity, Frames, and the ALE Method

Consider the bending of a beam in the general case of a large deformation (see Figure 2-24). In this case the deformation of the beam introduces an effect known as *geometric nonlinearity* into the equations of solid mechanics.

Figure 2-24 shows that as the beam deforms, the shape, orientation, and position of the element at its tip changes significantly. Each edge of the infinitesimal cube undergoes both a change in length and a rotation that depends on position. Additionally, the three edges of the cube are no longer orthogonal in the deformed

configuration (although typically for practical strains the effect of the nonorthogonality can be neglected in comparison to the rotation).

From a simulation perspective it is possible to solve the equations of solid mechanics on either a fixed domain (this is often called a *Total Lagrangian* formulation), or on a domain that changes continuously with the deformation. The latter approach is often called an *Updated Lagrangian* formulation. These two approaches also stand in contrast to the *Eulerian* formulation, which is often used for fluid mechanics. In an Eulerian formulation, the flow through a domain fixed in space is considered, while in the Lagrangian formulation, a fixed volume of material is considered.

In COMSOL Multiphysics, the concepts of a *material frame* and a *spatial frame* are used. Equations formulated in the material frame will give a Total Lagrangian formulation, while equations formulated in the spatial frame will give an Eulerian formulation.

See Frames and Coordinate Systems in the *Structural Mechanics Theory* chapter for more details.

Solid mechanics in COMSOL Multiphysics is formulated on the material frame. This is achieved by defining a displacement field for every point in the solid, usually with the components u, v, and w. At a given coordinate (X, Y, Z) in the reference configuration (on the left of Figure 2-24), the value of u describes the displacement of the point relative to its original position. The displacement is considered as a function of the material coordinates (X, Y, Z), but it is not explicitly a function of the spatial coordinates (x, y, z). The spatial coordinates give the current location in space of a point in the deformed solid. As a consequence, it is only possible to compute derivatives with respect to the material coordinates.

Taking derivatives of the displacement with respect to X, Y, and Z enables the definition of a strain tensor. There are possible representations of the deformation. Any reasonable representation must however be able to represent a rigid rotation of an unstrained body without producing any strain. The engineering strain fails here, thus it cannot be used for general geometrically nonlinear cases. One common choice for representing large strains is the *Green–Lagrange strain*. It contains derivatives of the displacements with respect to the original configuration. The values therefore represent strains in material directions. This choice allows a physical interpretation, but it must be realized that even for a uniaxial case, the Green–Lagrange strain is strongly nonlinear with respect to the displacement. If an object is stretched to twice its original

length, the Green–Lagrange strain is 1.5 in the stretching direction. If the object is compressed to half its length, the strain would read –0.375.

An even more fundamental quantity is the deformation gradient, which contains the derivatives of the deformed coordinates with respect to the original coordinates:

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \mathbf{u} + I$$

The deformation gradient contains all information about the local deformation in the solid, and can be used to form many other strain quantities. As an example, the Green–Lagrange strain is

$$\varepsilon = \frac{1}{2}(F^T F - I)$$

An element at a point (X, Y, Z) specified in the material frame moves with a single piece of material throughout a solid mechanics simulation. It is often convenient to define material properties in the material frame as these properties move and rotate naturally together with the volume element at the point at which they are defined as the simulation progresses. In Figure 2-24 this point is illustrated by the small cube highlighted at the end of the beam, which is stretched, translated, and rotated as the beam deforms. The three mutually perpendicular faces of the cube in the Lagrange

frame are no longer perpendicular in the deformed (spatial) frame. The deformed frame coordinates in this frame are denoted (x, y, z) in COMSOL.

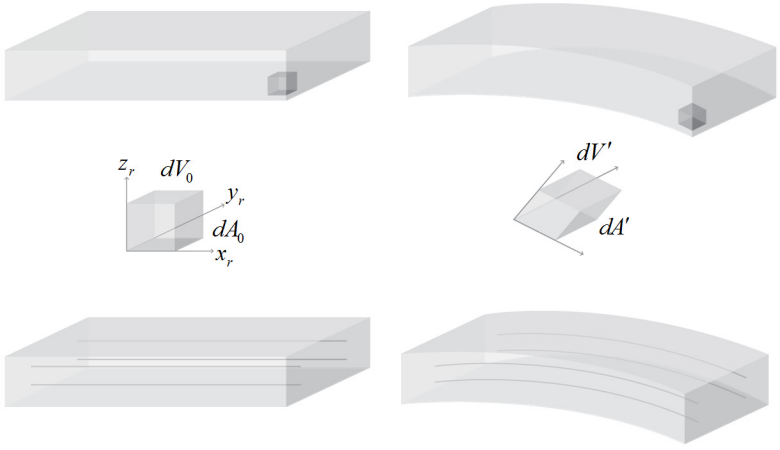


Figure 2-24: An example of the deformation of a beam showing the undeformed state (left) and the deformed state (right) of the beam itself with an element near its tip highlighted (top), of the element (center), and of lines parallel to the x-axis in the undeformed state (bottom).

It is important to note that, as the solid deforms, the Lagrangian frame becomes a nonorthogonal curvilinear coordinate system (see the lower part of Figure 2-24 to see the deformation of the X-axis). Particular care is therefore required when defining physics in this coordinate system.

FRAME CONTROL CONFLICTS

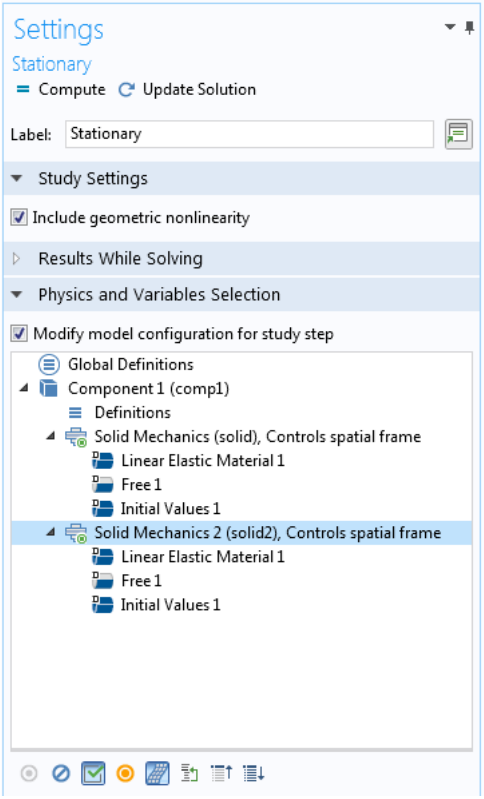
Under geometric nonlinearity, a Solid Mechanics interface (or a similar structural mechanics interface such as Multibody Dynamics) will assume control over the spatial frame in the domains where it is active. However, the definition of the spatial frame must be unique, and there may be conflicts. Some examples are:

- A Solid Mechanics interface and a moving mesh feature (for example, **Deforming Domain** or **Rotating Domain**) have a common selection. In this case, the selection in

the Solid Mechanics interface will be marked as “not applicable”. Thus, it is not possible to solve a structural mechanics problem on such a domain.

- There are more than one frame-controlling physics interfaces active on the same domain.

When two physics interfaces are competing for frame control, you will get the error message “Multiple moving frame specifications on the same selection” when trying to run the study. To identify the problem, go to the settings for the study step, and select **Modify model configuration for study step**. There, you will get an overview of the frame control in the model. Note that it is quite possible that several features control the spatial frame, as long as it is on different geometric selections.



When you select a physics interface in the tree view, you can click the **Control Frame Deformation** button ( ) to toggle whether that interface should control the spatial frame or not.

STRESS MEASURES

For example, in the spatial frame it is easy to define forces per reference area (known as *tractions*) that act within the solid and to define a stress tensor that represents all of these forces that act on a volume element. Such forces could be physically measured, for example, using an implanted piezoresistor. The stress tensor in the spatial frame is called the Cauchy or true stress tensor (in COMSOL Multiphysics this is referred to as the spatial stress tensor). To construct the stress tensor in the Lagrangian frame a tensor transformation must be performed on the *Cauchy stress*. This produces the *second Piola–Kirchhoff* (or material) stress, which can be used with the material strain to solve the solid mechanics problem in the (fixed) material frame. This is how the Solid Mechanics interface works when geometric nonlinearities are enabled.

For the Cauchy stress tensor, both the force components and the normal to the area on which the force is acting have fixed directions in space. This means that if a stressed body is subjected to a pure rotation, the actual values of the stress components will change. What was originally a uniaxial stress state might be transformed into a full tensor with both normal and shear stress components. In many cases, this is neither what you want to use nor what you would expect.

Consider for example an orthotropic material with fibers having a certain orientation. It is much more plausible that you want to see the stress in the fiber direction, even if the component is rotated. The Second Piola–Kirchhoff stress has this property as it is defined along the material directions. In the figure below, an originally straight cantilever beam has been subjected to bending by a pure moment at the tip. The xx-component of the Cauchy stress and Second Piola–Kirchhoff stress are shown. Since the stress is physically directed along the beam, the xx-component of the Cauchy stress (which is related to the global x direction) decreases with the deflection. The Second Piola–Kirchhoff stress, however, has the same through-thickness distribution all along the beam, even in the deformed configuration.

Figure 2-25: xx-components of the Cauchy stress tensor (top) and the second Piola–Kirchhoff stress tensor for an initially straight beam.

Another stress measure available in COMSOL Multiphysics is the First Piola–Kirchhoff stress. It is a multiaxial generalization of the nominal (or engineering) stress. The stress is defined as the force in the current configuration acting on the original area. The First Piola–Kirchhoff stress is an unsymmetric tensor, and is for that reason less attractive to work with. Sometimes you may also encounter the Kirchhoff stress, although it is not used in COMSOL Multiphysics. The Kirchhoff stress is just the Cauchy stress scaled by the volume change. It has little physical significance, but can be convenient in some mathematical and numerical operations.

Unfortunately, even without a rotation, the actual values of all these stress representations are not the same. All of them scale differently with respect to local volume changes and stretches. This is illustrated in the graph below. The xx-component of four different stress measures are plotted at the fixed end of the beam from the example above. At this point, the beam axis coincides with the x-axis, so the directions of all stress tensor components coincide. In the center of the beam, where strains, and thereby volume changes are small, all values approach each other. For a case with large rotation but small strains, the different stress representations can be seen as pure rotations of the same stress tensor.

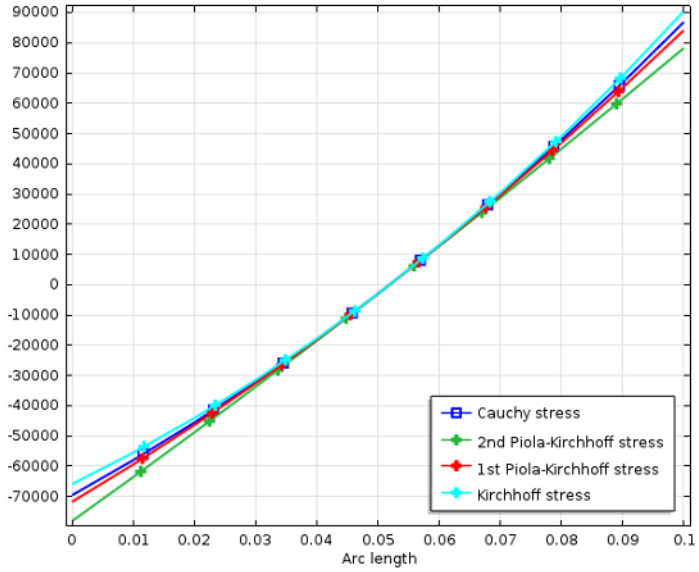


Figure 2-26: Stress distribution across the beam at the constrained end.

If you want to compute the resulting force or moment on a certain boundary based on the stresses, there are in practice only two possible choices: Either integrate the Cauchy stress over the deformed boundary, or integrate the First Piola–Kirchhoff stress over the same boundary in the undeformed configuration. In COMSOL Multiphysics this corresponds to selecting either **Spatial frame** or **Material frame** in the settings for the integration operator.

ALE METHOD

In the case of solid mechanics, the material and spatial frames are associated directly with the Lagrangian and Eulerian frames, respectively. In a more general case (for example, when tracking the deformation of a fluid, such as a volume of air surrounding a moving structure) tying the Lagrangian frame to the material frame becomes less desirable. Fluid must be able to flow both into and out of the computational domain, without taking the mesh with it. The *arbitrary Lagrangian-Eulerian* (ALE) method allows the material frame to be defined with a more general mapping to the spatial or Eulerian frame. In COMSOL Multiphysics, a separate equation is solved to produce this mapping — defined by the mesh smoothing method (Laplacian, Winslow, hyperelastic, or Yeoh) with boundary conditions that determine the limits of

deformation (these are usually determined by the physics of the system, whilst the domain level equations are typically being defined for numerical convenience). The ALE method offers significant advantages since the physical equations describing the system can be solved in a moving domain.

Deformed Geometry and Moving Mesh in the *COMSOL Multiphysics Reference Manual*

Geometric Nonlinearity for the Solid Mechanics Interface

In the settings for each material model in the Solid Mechanics interface, you can select a **Formulation** — **From study step** (default), **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain.

When **From study step** is selected, the study step controls the kinematics and the strain definition. In this situation, you can force a geometrically nonlinear analysis for a certain study step by selecting the **Include geometric nonlinearity** checkbox in the **Study Settings** section of the study step.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

If any active feature in the model requires the analysis to be geometrically nonlinear, the **Include geometric nonlinearity** checkbox in the study step is selected automatically, and it cannot be cleared. The following physics features force this behavior:

- Contact, because the deformation between the contacting boundaries is part of the solution.
- Moving mesh (when at least one deforming domain is active).
- Hyperelastic materials, which are always formulated for large strains.

For geometrically nonlinear analysis, the behavior of several features differs from the geometrically linear case:

- There is an important difference between using uppercase (X, Y, Z, R) and lowercase (x, y, z, r) coordinates in expressions. The lowercase coordinates represent the deformed position, and this introduces a dependency on the solution.
- Many features, such as coupling operators, can be specified as operating in either the material (X, Y, Z) or the spatial (x, y, z) frame. This setting does not make a

difference unless a geometrically nonlinear analysis is performed. In most cases you would want to do the operation in the material frame.

- The strain representation changes from using engineering strains to Green–Lagrange strains, unless **Geometrically linear** is selected in the **Formulation** list of a certain material.
- Where Green–Lagrange strains are used, Second Piola–Kirchhoff stresses are also used. This means that material data must be given in terms of these quantities. This distinction is important only when the strains are actually large.
- For each material model, the **Strain decomposition** list determines how inelastic deformations are treated. It is possible to choose an additive decomposition of strains or a multiplicative decomposition of deformation gradients. This option is not available when the formulation is set to **Geometrically linear**, for which an additive strain decomposition is used.
- Pressure loads are interpreted as follower loads, so that the direction of the load as well as the loaded area depend on deformation.
- Rigid connectors take finite rotations into account.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

Geometric Nonlinearity for the Shell, Plate, Membrane, Beam, and Truss Interfaces

For the Shell, Plate, Membrane, and Truss interfaces, geometrically nonlinear analysis is handled in the same way described above. For the Membrane and Truss interfaces, geometric nonlinearity must almost always be used, because nonlinear effects supply stiffness in the transverse direction. The geometric nonlinearity in the Beam interface is limited to large rotations and displacements, but small strains. It is possible to force a geometrically linear formulation in the beam interface.

Solving Geometrically Nonlinear Problems

Depending on the geometrically nonlinear effects that appear in your model, you may have to use different solution strategies. Some problems in this class are strongly nonlinear, while others show only a weak deviation from linearity. Some guidelines are:

- If the problem has a path dependent solution, then it must be solved in an incremental way in order to give a correct solution. Problems including for example plasticity or friction belong to this class. If you do the analysis in time domain, then the solution is inherently incremental. If the analysis is stationary, invoke the parametric continuation solver by adding an **Auxiliary Sweep**, and ramp up some loading parameter. In either case, make sure that the step size is not too large.
- Problem that have a unique solution, like an elastic model subjected to large rotations or strains can be solved in a single static load step without loss of accuracy. It is however possible that such an approach will not converge, in which case a parametric continuation solver must be used.
- In problems involving large rotations, the default settings of the nonlinear solver will sometimes give a solution strategy that is too conservative. You can often decrease the solution time significantly by modifying the settings under **Method and Termination** in the settings for the **Fully Coupled** or **Segregated Step** node in the solver sequence. Set **Nonlinear method** to **Constant (Newton)** and use a rather high **Damping factor**. In most cases the value 1 will work.
- If you model a situation which to a large extent is a rigid rotation, it is often necessary to use tighter tolerances than usual in order to avoid spurious stresses. Since the strains are computed from the differences of the displacements in an element, even a small relative error in the displacements (which are large) can cause significant strains. This will be visible in a case where the actual stresses are small.

Prestressed Structures

You can analyze eigenfrequency, frequency domain, or time-dependent problems where the dynamic properties of the structure are affected by a preload, such as a tensioned string.

Usually, a study of a prestressed problem includes using two study steps. The first step is a **Stationary** step in which the static preload is applied. The preload step can be computed with or without taking geometric nonlinearity into account. In the second study step, where you compute the eigenfrequency or the frequency response, it is

necessary to take geometric nonlinearity into account. Even if the displacements and strains are small, this is what gives the prestress contribution to the equations.

There are three predefined study sequences for prestressed dynamic analysis:

- Eigenfrequency, Prestressed
- Frequency Domain, Prestressed
- Frequency Domain, Prestressed, Modal

The prestressed study types assume that the loading causes small perturbations around the prestressed state.

In the predefined study sequences, the **Include geometric nonlinearity** checkbox is selected by default also in the stationary prestress study step.

In many cases, a geometrically linear assumption is sufficient for this step. If no other nonlinearities are present in the prestress step, you can improve performance by instead using a geometrically linear formulation, thus avoiding unnecessary iterations.

To understand which option to use, it is important to understand how the two study steps are linked. It is the displacement field of the first study step that is used for computing the stresses including stress stiffening effects in the second study step. Thus, it does not directly matter whether the stresses in the first study step are right or wrong. The only thing that matters is that when you switch from a geometrically linear to a geometrically nonlinear formulation, the stresses that you get with the nonlinear formulation are correct enough, using the given displacement field.

In a general nonlinear analysis, like a **Stationary** or **Time Dependent** study step, the full combined effect of the prestress and other loads will automatically be included.

The same principles apply also to a linear buckling analysis, except that both study steps should be geometrically linear. The nonlinear contribution is included in the formulation of the buckling eigenvalue itself.

FOLLOWER LOADS

Loads that change orientation with deformation, such as a pressure, actually contribute not only to the load but also to the stiffness. This is a physical effect and not just a numerical artifact. Whether such loads are included or not in an **Eigenfrequency** study

step will affect the computed eigenfrequencies. If you for some reason do not want this effect, you must suppress the load in the **Physics and Variables** section of the **Eigenfrequency** node.

If you use a local coordinate system for describing a load, you must, in case of geometric nonlinearity, pay attention to whether that coordinate system has constant axis orientations or not. As an example, the default boundary system has **Frame** set to **Deformed Configuration**, so that a load represented in that system will behave as a follower load. Change to **Reference Configuration** if the load should act in fixed directions.

In more general terms, any feature in which there is a dependency on the choice of frame (material or spatial) can potentially affect the outcome of a prestressed analysis.

CONTACT AND PRESTRESSED ANALYSIS

If a contact is included in the prestress load case, you can perform a subsequent eigenfrequency or frequency-domain analysis in which the linearization will be made around the computed contact state. The connection between the two boundaries is elastic, using the penalty factor as stiffness, both for the penalty method and the augmented Lagrangian method. If the augmented Lagrangian method is used for the contact modeling, then the penalty factor must be constant and not dependent on the iteration number.

The contact state is considered as fixed, so there is no harmonic perturbation in contact quantities, such as the contact pressure.

The gap distance in a contact analysis is computed from the coordinates of the contacting boundaries in the spatial frame. The spatial frame is defined by the displacements in the preload step and is not affected by the displacements in a subsequent perturbation step.

INELASTIC STRESSES AND STRAINS

When inelastic stresses and strains are part of the problem description, you need to make some extra considerations. Such contributions are formally part of the constitutive model but generate load vector contributions.

There are three **Preset** study types which can be used to set up these two study steps: **Eigenfrequency, Prestressed**; **Frequency Domain, Prestressed**; and **Linear Buckling**.

If you want to explicitly prescribe the stress field for a prestressed analysis rather than solving for it, you should not use the two study step procedure. In such a case, prescribe the stress field using an **Initial Stress and Strain**, **External Stress**, or **External Strain** node. Then add a separate **Eigenfrequency** or **Frequency Domain** study and select **Include Geometric Nonlinearity** in the settings for the study step.

Eigenfrequency, Prestressed, Frequency Domain, Prestressed, and Linear Buckling in the *COMSOL Multiphysics Reference Manual*

- For an example of a general prestressed eigenfrequency analysis see *Bracket — Eigenfrequency Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_eigenfrequency**.
- For an example of an analysis where the stress state is explicitly prescribed, see *Vibrating Membrane*: Application Library path **Structural_Mechanics_Module/Verification_Examples/vibrating_membrane**.

Geometric Nonlinearity for the Piezoelectric Material

PIEZOELECTRIC MATERIALS WITH LARGE DEFORMATIONS

The linear piezoelectric equations as presented in About Piezoelectric Materials with engineering strains are valid if the model undergoes only relatively small deformations. As soon as the model contains larger displacements or rotations, these equations produce spurious strains that result in an incorrect solution. To overcome this problem, so-called large deformation piezoelectric equations are required.

The Piezoelectric Material implements the large deformation piezoelectric equations according to Yang (Ref. 1). Key items of this formulation are:

- The strains are calculated as the Green–Lagrange strains, ε_{ij}:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \sum_k \frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}\right) \tag{2-18}$$

  Green–Lagrange strains are defined with reference to an undeformed geometry. Hence, they represent a Lagrangian description. In a small-strain, large rotational analysis, the Green–Lagrange strain corresponds to the engineering strain in directions that follow the deformed body.
- Electrical field variables are calculated in the material directions, and the electric displacement relation is replaced by an expression that produce electric polarization in the material orientation of the solid.
- In the variational formulation, the electric energy is split into two parts: The polarization energy within the solid and the electric energy of free space occupied by the deformed solid.

The first two items above result in another set of constitutive equations for large deformation piezoelectricity:

$$S = c_E\varepsilon - e^T\mathbf{E}_\mathrm{m}$$
$$\mathbf{P}_\mathrm{m} = e\varepsilon + \varepsilon_0(\varepsilon_{rS} - \mathbf{I})\mathbf{E}_\mathrm{m}$$

where S is the second Piola–Kirchhoff stress; ε is the Green–Lagrange strain, $\mathbf{E}_\mathrm{m}$ and $\mathbf{P}_\mathrm{m}$ are the electric field and electric polarization in the material orientation; $\mathbf{I}$ is the identity matrix; and c_E, e, and ε_{rS} are the piezoelectric material constants. The expression within parentheses equals the dielectric susceptibility of the solid:

$$\chi = \varepsilon_{rS} - \mathbf{I}$$

The electric displacement field in the material orientation results from the following relation

$$\mathbf{D}_\mathrm{m} = \mathbf{P}_\mathrm{m} + \varepsilon_0 JC^{-1}\mathbf{E}_\mathrm{m}$$

where C is the right Cauchy–Green tensor

$$C = F^T F$$

Fields in the global orientation result from the following transformation rules:

$$\begin{aligned}\mathbf{E} &= F^{-T}\mathbf{E}_\mathrm{m}\\ \mathbf{P} &= J^{-1}F\mathbf{P}_\mathrm{m}\\ \mathbf{D} &= J^{-1}F\mathbf{D}_\mathrm{m}\\ \rho_\mathrm{v} &= \rho_\mathrm{V}J^{-1}\end{aligned} \tag{2-19}$$

where F is the deformation gradient; J is the determinant of F; and ρ_v and ρ_V are the volume charge density in spatial and material coordinates, respectively. The deformation gradient is defined as the gradient of the present position of a material point $\mathbf{x} = \mathbf{X} + \mathbf{u}$:

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$$

Finally, one can rewrite the constitutive equations as

$$\begin{aligned}S &= c_E\varepsilon - e^T\mathbf{E}_\mathrm{m}\\ \mathbf{D}_\mathrm{m} &= e\varepsilon + \varepsilon_0\varepsilon_{rS}\mathbf{E}_\mathrm{m} + \varepsilon_0(JC^{-1} - \mathbf{I})\mathbf{E}_\mathrm{m}\end{aligned}$$

DECOUPLED MATERIALS WITH LARGE DEFORMATIONS

The large deformation formulation described in the previous section applies directly to materials not being piezoelectric if the coupling term is set to zero: $e = 0$. In that case, the structural part corresponds to the large deformation formulation described for the solid mechanics interfaces.

The electrical part separates into two different cases: For solid domains, the electric energy consists of polarization energy within the solid and the electric energy of free space occupied by the deformed solid — the same as for the piezoelectric materials. For nonsolid domains this separation does not occur, and the electric displacement in these domains directly results from the electric field — the electric displacement relation:

$$\mathbf{D} = \varepsilon_0\varepsilon_\mathrm{r}\mathbf{E}$$

The Electrostatics Interface *in the COMSOL Multiphysics Reference Manual*

LARGE DEFORMATION AND DEFORMED MESH

The Piezoelectricity, Solid Interface can be coupled with the Moving Mesh (ALE) interface in a way so that the electrical degrees of freedom are solved in an ALE frame. This feature is intended to be used in applications where a model contains nonsolid domains, such as modeling of electrostatically actuated structures. This functionality is not required for modeling of piezoelectric or other solid materials.

The use of ALE has impacts on the formulation of the electrical large deformation equations:

- The first impact is that with ALE, the gradient of the electric potential directly results in the electric field in the global orientation, and the material electric field results after transformation.
- The most visible impact is on the boundary conditions. With ALE, any surface charge density or electric displacement is defined per the present deformed boundary area, whereas for the case without ALE, they are defined per the undeformed reference area.

Deformed Geometry and Moving Mesh *in the COMSOL Multiphysics Reference Manual*

References

1. J. Yang, *An Introduction to the Theory of Piezoelectricity*, Springer Science and Business Media, N.Y., 2005.

2. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, Boca Raton, FL, 2010 (solidmechanics.org).

Contact Modeling

Overview

Situations where objects come into contact with each other occur frequently in mechanical simulations. Setting up and solving such contact problems can be a challenging task, and this section contains information about important aspects of creating models involving contact.

In a contact analysis, you solve for the contact state and the contact forces. The most fundamental quantity in a contact analysis is the distance between the objects which may come into contact, the *gap*. If the gap is positive then there is no interaction. The task of the contact condition is to ensure that the gap never becomes negative; that is, to avoid *overclosure*. In order to accomplish this task, contact forces must be introduced.

For normal contact, the state only consists of being in contact or not, and the force variable is the contact pressure in the normal direction. With friction included, there are two possible states for the relative tangential displacement when in contact: *sticking* or *sliding*. The tangential force vector is added as force variable.

The contact analysis functionality in COMSOL Multiphysics also includes the possibility to prescribe adhesion and decohesion between the contacting objects, and to model removal of material by wear when the objects are sliding relative to each other.

In this section:

- Setting Up a Contact Problem
- Contact Pairs
- General Contact Pairs
- Meshing for Contact Analysis
- Settings for Contact Nodes
- Settings for General Contact Nodes
- Quasistatic Contact Analysis
- Implicit Dynamic Contact Analysis
- Explicit Dynamic Contact Analysis
- Multiphysics Contact Analysis
- Special Types of Contact Problems
- Fallback Conditions to Contact Regions
- Solver Settings for Contact Analysis
- Monitoring the Solution
- Dependent Variables in Contact Analysis
- Important Contact Variables

Setting Up a Contact Problem

Mechanical contact can be modeled between boundaries in the Solid Mechanics; Solid Mechanics, Explicit Dynamics; Multibody Dynamics; Shell; Layered Shell; and Membrane interfaces. You can model contact not only within a single physics interface, but also between two physics interfaces, or even between a physics interface and any boundary having a mesh, using a combination of **Contact** and **General Contact** nodes.

The **General Contact** node is available in the Solid Mechanics; Solid Mechanics, Explicit Dynamics; and Multibody Dynamics interfaces.

Within the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces, you can either use the more general formulations available through the **Contact** and **General Contact** nodes, or you can use a simplified version called **Interior Contact**.

USING CONTACT PAIRS AND THE CONTACT NODE

To model a mechanical contact problem using the **Contact** node, you must do the following fundamental steps:

- In the finalization step of the geometry sequence, you should normally have **Action** set to **Form an assembly**. If **Form a union** is used, then the contacting boundaries must be geometrically separated in the initial configuration.
- Add a **Contact Pair** ( ) under **Definitions**. A contact pair defines a mapping between two sets of boundaries; the *source* and *destination* boundaries. Contact pairs can also be added automatically, based on boundary adjacency when the **Form an assembly** action is used.
- Use the default **Contact** node or add new **Contact** nodes in the physics interface. In the **Contact** node, you select the contact pairs to be used, and provide the settings for the physical and numerical properties of the contact model.
- If relevant, add **Friction**, **Slip Velocity**, **Adhesion**, **Decohesion**, or **Wear** subnodes to **Contact**.

- Select a suitable study type. You can analyze contact using the following types of study steps:
  - Stationary; commonly using an auxiliary sweep.
  - Bolt Pretension
  - Time Dependent
  - Explicit Dynamics
  - Any type of prestressed analysis. In this case, the full contact problem is solved in the preload step, and in the subsequent perturbation step the contact state is considered as fixed.

The fact that you add a **Contact** node to your model will automatically make all study steps geometrically nonlinear. For the default **Contact** node, this requirement can be removed by selecting **Disconnect pair**.

Because of the multiphysics capabilities of COMSOL Multiphysics, the setup of a contact problem is split into two parts. The definition of the contact pair is made under **Definitions**, and can be shared between several physics interfaces. This part of the contact problem defines the geometric properties of the contact, such as search and mapping operations between the selected boundaries. The physics-related definitions are then made in the respective physics interfaces.

When to Use the Contact Pair Formulation

The contact pair provides a general formulation that can handle arbitrarily large deformations including finite sliding with or without friction. For cases where you know which boundaries may interact through contact when setting up the model, it is often the preferred option. It limits the size of the contact problem to selected boundaries, thus keeping down the computational cost and complexity.

The **Contact** node includes a wide range of contact methods ranging from the lean and robust but approximate penalty method, to the more accurate but expensive augmented Lagrangian and Nitsche methods. The latter methods are not supported by **General Contact**. Thus, for contact interactions where high accuracy is important, use the **Contact** node.

Complex contact interactions involving **Adhesion**, **Decohesion**, and **Wear** can only be modeled using the **Contact** node.

In a multiphysics analysis, the contact pair formulation can also incorporate for example changes in the heat flux or electric current through the contacting boundaries. You will then also need to add corresponding features in the other participating interfaces, like a **Thermal Contact** node in the Heat Transfer in Solids interface. The contact state and contact pressure used by other physics interfaces are always supplied by the structural mechanics interface

- Documentation of the Contact, Friction, Slip Velocity, Adhesion, Decohesion, and Wear nodes.
- Contact Analysis Theory.
- Identity, Contact, and Sector Symmetry Pairs in the *COMSOL Multiphysics Reference Manual*.

USING GENERAL CONTACT PAIRS AND THE GENERAL CONTACT NODE

To model a mechanical contact problem using the **General Contact** node, you must do the following fundamental steps:

- Add a **General Contact Pair** ( ) under **Definitions**. By default, a general contact pair defines a mapping between all geometric objects in the model, including domains and free boundaries. When all objects are included, a single general contact pair is sufficient.
- Use the default **General Contact** node or add new **General Contact** nodes in the physics interface. In the **General Contact** node, you select the general contact pairs to be used.
- The main settings for the physical and numerical properties of the contact model are made in the **Contact Model** subnode. Add more **Contact Model** nodes to use different settings for different boundaries. Geometric properties of boundaries can be adjusted by adding **Offset** subnodes.

- If relevant, add **Friction** or **Damping** subnodes to **Contact Model**.
- Select a suitable study type. You can analyze contact using the following types of study steps:
  - Stationary; commonly using an auxiliary sweep.
  - Bolt Pretension
  - Time Dependent
  - Explicit Dynamics
  - Any type of prestressed analysis. In this case, the full contact problem is solved in the preload step, and in the subsequent perturbation step the contact state is considered as fixed.

The fact that you add a **General Contact** node to your model will automatically make all study steps geometrically nonlinear. For the default **General Contact** node, this effect can be removed by selecting **Disconnect pair**.

When to Use the General Contact Pair Formulation

The general contact pair provides a general formulation that can handle arbitrarily large deformations including finite sliding with or without friction. Compared to the contact pair formulation, contact detection is automatic with no need to distinguish between source and destination boundaries, making it easy to set up the contact problem even for large and complex assemblies. Special contact situations, such as self-contact, are also intrinsically supported with no extra modeling steps required.

A typical usage of the general contact pair is to let it define the overall contact interaction for the entire geometry, while other pairs define interactions for specific boundaries where more detail is required. For example, add a contact pair to use the more accurate augmented Lagrangian method in a bolted connection, or add an identity pair to model continuity between a set of boundaries.

- Documentation of the General Contact, Contact Model, Offset, Damping (General Contact), and Friction (General Contact)nodes.
- Contact Analysis Theory.
- Identity, Contact, and Sector Symmetry Pairs in the *COMSOL Multiphysics Reference Manual.*

USING THE INTERIOR CONTACT NODE

To model a mechanical contact problem using the **Interior Contact** node, you must do the following fundamental steps:

- Add an **Interior Contact** node.
- Select the set of boundaries that form the region where contact is defined.
- Provide the settings for the physical and numerical properties of the contact model.
- If relevant, add **Friction**, **Adhesion**, or **Decohesion** subnodes.
- Select a suitable study type. You can analyze contact using the following types of study steps:
  - Stationary; commonly using an auxiliary sweep.
  - Bolt Pretension
  - Time Dependent
  - Explicit Dynamics
  - Any type of prestressed analysis. In this case, the full contact problem is solved in the preload step, and in the subsequent perturbation step the contact state is considered as fixed.

When to Use the Interior Contact Formulation

The interior contact formulation does not require the addition of any contact pairs. It is applied as a *slit condition* on an interior boundary in a solid. A slit condition keeps the mesh, but allows different displacements on the elements on the two sides of the boundary. The computationally expensive contact search is avoided, instead the gap and sliding distances are measured as the difference in displacements across the slit. This also has the advantage that **Interior Contact** does not force a geometrically nonlinear study.

The interior contact formulation is therefore useful when the sliding between contact surfaces is small, typically limited to a small fraction of the mesh element size. An example of when such a behavior is expected is in bolted connections. Also, many decohesion problems fall in this category.

Interior Contact does not have any built-in multiphysics capabilities, but it is possible extend it using equation-based modeling.

To a large extent, **Interior Contact** has the same settings and properties as the **Contact** node. However, the **Wear**, **Slip Velocity**, and **Stabilization** subnodes are not available.

Because of the inherent limitations, the interior contact formulation cannot be used for modeling dynamics events such as impact.

- Documentation of the Interior Contact, Friction, Adhesion, and Decohesion nodes.
- Contact Analysis Theory.

INCLUDING FRICTION

In real life, there is always some friction between contacting objects, but this is sometimes ignored when setting up the mechanical contact problem. There are several reasons to do this simplification:

- In many cases, only the normal forces are significant for the general force distribution in the structure, while the frictional forces only cause minor local effects.
- The values of the friction coefficients are difficult to obtain, and unless the structure is assembled under well controlled conditions, the magnitude of the friction can vary a lot. If there is a large degree in uncertainty in the input data, it can be argued that the inclusion of friction does not add much to the quality of the results.
- Adding friction to a contact problem will often increase the computation time significantly, or even cause convergence problems.

There are a number of situations when friction modeling cannot be avoided. Some of them are:

- When a significant portion of the load is carried by shear tractions acting on the contact boundaries.
- When a tangential force is necessary to maintain stability and to avoid rigid body motions. In many cases, it is however possible to replace the friction by a suitable boundary condition instead, as long as there are no resultant forces being resisted by such a constraint.
- When modeling wear.
- When the frictional dissipation is an important component of a dynamics problem, or when it is needed as a heat source in a thermal analysis.

In some cases, such as when a brake pad slides along a brake disc, the size and orientation of the slip velocity is known. You can then employ a simplified form of friction modeling by assuming the tangential contact to always be in a slip state, which

simplifies the computation of the friction forces. This is done using the Slip Velocity node. This is particularly useful for wear modeling.

INCLUDING ADHESION AND DECOHESION

You can also specify that the contacting boundaries stick to each other, so that they will not separate or slide. The onset of adhesion, when the boundaries become permanently attached to each other, can be based on several criteria:

- When a certain contact pressure is exceeded.
- When the gap distance between the contact boundaries is smaller than a certain value.
- When a user specified logical expression is fulfilled. This can for example be used if an adhesive cures when a certain temperature is exceeded.
- From the start of the analysis. This case is particularly useful if you are interested in modeling the subsequent tearing of a thin glue layer by decohesion.

If adhesion is active between the contact boundaries, it is possible to break the bond by adding a decohesion model. You can choose between several different decohesion models.

Adhesion and friction can be combined, but during the time that two boundaries are bonded to each other through adhesion, any settings for friction are ignored.

INCLUDING WEAR

It is possible to model adhesive or abrasive wear of the material when the contacting boundaries are sliding along each other. The removal of material from the domain adjacent to the contact boundary can be modeled using three, fundamentally different, approaches.

For the Solid Mechanics and the Multibody Dynamics interfaces, the most general approach relies on the deformed geometry concept, where material is actually removed by using an adaptive mesh technique. In the second, simplified, approach wear is incorporated as an offset in the contact condition. This approach is computationally less expensive, and is suitable as long as only small amounts of material are removed, and wear does not change the orientation of the normal to the boundary significantly. In the Shell and Membrane interfaces the structural domain is represented by only a meshed surface. Therefore, a different, more suitable approach is used, in which the thickness variable and the midsurface offset to the meshed boundary are modified.

Computing the amount of wear involves solving a rate equation, hence, it is only possible to compute wear in time dependent study steps. The wear rate is typically a

function of the contact pressure and the relative slip velocity between the contacting boundaries.

SELECTING THE CONTACT METHOD

In COMSOL Multiphysics, there are three classes of methods available for implementing the contact conditions: the *penalty* method, the *augmented Lagrangian* method, and the *Nitsche* method.

The Penalty Method

The default penalty method is a simple and robust method to introduce the contact conditions. Roughly speaking, it is based on inserting a stiff distributed spring, active only in compression, between the contacting boundaries. In addition to the robustness, it has the advantage that no special solver is required, which makes it easier to set up multiphysics problems and time-dependent studies. However, the penalty method only enforces the contact conditions approximately. The contact forces computed are thus less accurate than when using, for example, the augmented Lagrangian method, and there is always some overclosure between the contacting surfaces. For stick friction, there is also an "elastic" deformation due to the penalization of the constraint.

When using the penalty method, there is always a trade-off between accuracy and stability. While a large penalty factor will reduce unphysical overclosures, the problem may become ill-conditioned and unstable if it is too large. It might therefore be beneficial to accept some penetration between the contacting objects. Note, however, that if the penalty factor is too small, the contact condition may be violated.

Damping can be added when using the penalty method for transient problems. For the **Contact** node, damping is enabled by selecting the **Penalty, dynamic** method; while for the **General contact** node, it is enabled by adding a **Damping** subnode to **Contact Model**. Using the same analogy as above, contact damping introduces a distributed viscous damper that is only active when the gap is negative. Moreover, the viscous damper is unidirectional and only applies a force to negative gap velocities.

It is possible to either add the viscous damper to the spring or let the viscous damper replace the spring. The latter implies a purely viscous formulation that requires the duration of the contact should be small, otherwise the overclosure will eventually grow and violate the nonpenetration condition.

The Augmented Lagrangian Method

The augmented Lagrangian method provides better accuracy, but at a higher computational cost, and is often less stable from a convergence point of view. All

contact conditions are enforced in a weak or integral sense, and thus evaluated in the integration points on the destination boundary. For normal contact, it is thus possible for a node to have a small penetration into the source boundary, even for a well converged solution. Both, the contact pressure and the friction forces are added as extra degrees of freedom to enforce the contact constraints.

When using the augmented Lagrangian method, it is possible to choose between a segregated and a coupled solution method. The segregated solution method sets up a special type of segregated solver sequence, where the extra degrees of freedom related to contact are updated in a separate segregated solver step. As its name implies, there is no such need when using the coupled solution method, and you can more freely set up the solver sequence. This less restrictive solver requirement can be particularly useful for multiphysics problems where the penalty method does not provide sufficient accuracy.

The segregated augmented Lagrangian method is additionally available in a specialized formulation intended for dynamic contact problems. It is based on a viscous formulation that for the normal contact constrains the rate of the gap to be zero, rather than the gap. Since the formulation is viscous, the duration of the contact should be small, otherwise the overclosure will eventually grow and violate the nonpenetration condition.

The Nitsche Method

The Nitsche method for contact is an extension of the general method suggested by J. Nitsche in 1971 to weakly impose Dirichlet conditions. Conceptually, it can be viewed as an enhancement of the penalty method where the surface traction of the adjacent domains is used to improve the accuracy of the contact condition. Also, as for the penalty method, it has the advantage that no special solver is required, which makes it easier to set up multiphysics problems and time-dependent studies.

The Nitsche method is intended as an alternative to the augmented Lagrangian method for problems where the penalty method lacks sufficient accuracy. Relevant examples where this might be the case includes:

- Large deformation problems with hyperelasticity
- Self-contact
- Problems where the structural stiffness of the global system is of importance

For the first two problems in particular, the accuracy and stability can be further improved by increasing the quadrature order for the contact equations, see also

Quadrature Settings. Hence, the Nitsche method uses a higher quadrature order by default than the other contact methods.

COMSOL Multiphysics supports three different formulations of the Nitsche method:

- Symmetric
- Skew-symmetric
- Incomplete

The default incomplete formulation is recommended for the majority of cases and provides a good trade-off between performance and stability. In cases where the default formulation fails, the skew-symmetric formulation can be an alternative, but it includes additional equations that are expensive to evaluate that can increase the solution time.

The Nitsche method is available in the Solid Mechanics; Solid Mechanics, Explicit Dynamics; and Multibody Dynamics interfaces.

CONTACT DETECTION

A contact mapping is made from a source boundary to a destination boundary. The mapping is established from a point on the destination boundary by making a contact search to find the closest mesh element of the source boundary in the direction of the contact normal of the destination. For a source mesh element to be accepted as a valid contact mapping, it must be positioned in front of the destination. That is, the contact normal of the source must point toward the contact normal of the destination.

The opposite contact mapping, destination to source, is also possible but is seldom used for mechanical contact.

Generally, the contact normal coincides with the geometry normal that is determined by the direction of the boundary. However, there are situations where this is not the

case, or where the **Contact** or **General Contact** nodes can control the sign of the contact normal. Some common situations are summarized in the following:

- If the contact boundary is on the exterior of a domain, then the geometry normal always point outward from the domain. This is the most common case, and then no other considerations are needed.
- In some cases, like Fluid–Structure interaction, there may be a domain with a moving mesh between the two contacting objects. The source and destination boundaries will then, in the geometrical sense, be interior boundaries. In this case, the physics interface (for example Solid Mechanics) ensures that the contact normal is pointing outward from the boundaries that are external to the physics interface. Thus, this case is also handled automatically.
- When a boundary without an adjacent domain is selected, you need to be careful so that the normal is pointing in the intended direction. For this case it is possible to change the sign of the contact normal by using the Reverse Normal subnode of the **Boundary System** node
- When using the Shell, Layered Shell or Membrane interfaces, contact can potentially occur on both sides of the boundary. In a single **Contact** node, you can only model contact on one side. In the **Contact Surface** section, you can select whether the contact should occur on the top side or bottom side. The 'top' and 'bottom' sides are defined by the orientation of the physics interface normal, which may differ from the geometry normal. In these interfaces, the orientation of the physics normal is controlled by the Boundary System that is attached to each boundary through the material models. The normal direction can be reversed using the settings in the **Boundary System** node.

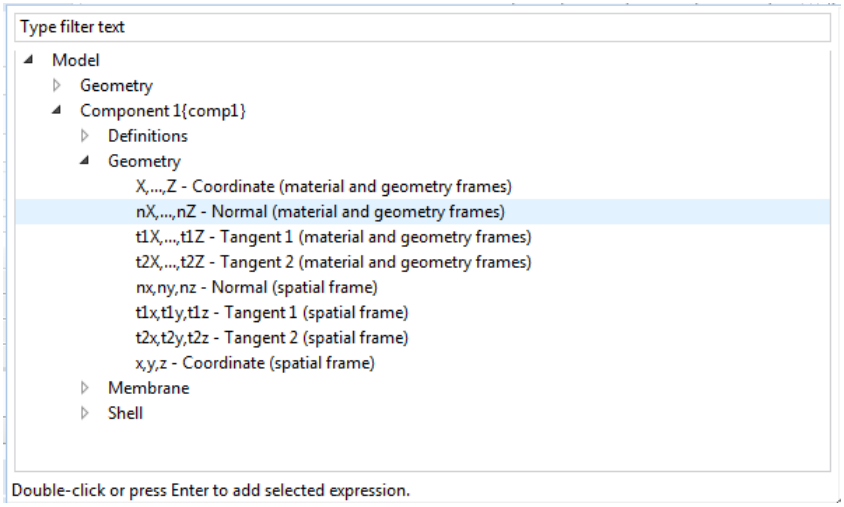
The contact normal can be visualized by plotting `<pair_tag>.n<coord_label>`. For example, the variable `p1.nx` gives the x-component of the spatial normal used by contact pair `p1`. Plotting the contact normal vector can be useful to verify that contact normal for a source and destination point toward each other.

Visualizing the Orientation of Boundaries

In 2D, a boundary is represented by a line. If you follow the line from its start point to its endpoint, the positive normal points to the left. The line orientations can be visualized by selecting the **Show edge direction arrows** checkbox in the appropriate **View** node under **Definitions**.

In 3D, the rules for the orientation of a boundary are more complicated. In general, you have to visualize the normals, in order to check its orientation. This can be done in different ways:

- For a mesh without physics, you will need to use the general result presentation tools. Follow these steps:
  - Run **Get Initial Values** for an arbitrary study in order to create data for result presentation.
  - Add a **3D Plot Group** under **Results**.
  - Add an **Arrow Surface** plot to the new plot group.
  - In the **Replace Expression** ( ) dialog, select the geometry normal.



- When working with the Shell or Membrane interfaces, select **Enable physics symbols** in the settings for the interface. You will then see the physics normals plotted if you select a material model like **Linear Elastic Material** in the Model Builder.

CONTACT BETWEEN PHYSICS INTERFACES

Contact can seamlessly be modeled between all applicable structural mechanics interfaces. Equations are only added by **Contact** and **General Contact** nodes with a selection that intersects the destination boundaries of a pair. Some properties, such as contact surface offset, can, however, be added as long as either a source or a destination boundary is applicable. **Contact** nodes in physics interfaces defined on boundaries, for

example in the Shell interface, automatically account for the assigned thickness on both source and destination boundaries.

SOURCE SELECTION OUTSIDE PHYSICS INTERFACES

In most cases, the source and destination boundaries belong to the same physics interface. However, the source side only needs to have a mesh and can optionally have one or more physics interfaces attached to it.

If the source boundary is not part of a physics interface with a **Contact** or **General contact** node, the contact mapping uses only the current location of the mesh, ignoring any physical properties that may exist there. In this case, the physics has no control of the contact normal used by the source boundary. Care must therefore be taken to ensure that the contact normal of the source and destination points toward each other for the contact detection to work.

There are two main scenarios where you may want to use a source selection that is not in the current physics interface:

Fixed Rigid Wall

If one side of a contact pair can be considered as rigid and fixed in space, then it is sufficient to add a meshed boundary at that location without any physics.

Moving Rigid Wall

This case is similar to the previous. In order to prescribe the path of the rigid wall, add **Moving Mesh** with a **Prescribed Deformation** node under **Definitions**.

Since the source boundary only requires a mesh, Shell, Layered Shell, and Membrane interfaces are in fact automatically included in the contact analysis when a general contact pair is used, even though they do not support the **General Contact** node. However, contact is detected only with the meshed boundary and only in the direction of the contact normal.

Contact Pairs

Contact pairs ( ) are only used by **Contact** nodes.

SELECTING SOURCE AND DESTINATION

To decide which boundaries to assign as source and destination in a contact pair, consider the following guidelines:

- If there is a significant difference in the stiffness in the normal direction between the source and destination boundaries, select the stiffer side as source. This is especially important if the difference in stiffness is quite large (for example, more than a factor of ten).
- When the contacting parts have approximately the same stiffness, consider the geometry of the boundaries instead. Make a concave boundary the source and a convex boundary the destination rather than the opposite.
- If one of the boundaries belongs to a part that is rigid it should be selected as the source boundary. Rigid boundaries can be created in several ways, for example
  - The underlying domain has the **Rigid Material** material model.
  - The boundary or its underlying domain is constrained by, for example, **Fixed Constraint** or **Prescribed Displacement**.
  - The source boundary is meshed, but has no physics attached.
  - The source boundary position is controlled by **Prescribed Deformation** under **Moving Mesh**.
- If only one side of the contact pair is within a physics interface that has a **Contact** node, that side must be the destination side.

For efficiency, include only those boundaries that can potentially come in contact in the destination selection. All equations are formed on the destination boundary, and includes performing the contact search which can be expensive.

If it is difficult to follow the above guidelines, the same boundaries can be selected as both source and destination. Doing so results in an unbiased (or symmetric) formulation that is less sensitive to, for example, difference in stiffness or mesh density between the contacting boundaries. However, such a formulation involves evaluating equation and contact mapping at additional points and can thus be more expensive to use.

STRATEGIES FOR SELECTING CONTACT PAIRS

A **Contact** node can reference any number of contact pairs.

A contact pair is just a definition, and does not have to be referenced by any **Contact** node.

Neither the source, nor the destination, in a contact pair has to be geometrically contiguous. In practice, this means that you often only need a few contact pairs in a model. The number of pairs actually needed will be determined by how many different settings that are required in the **Contact Pair** and **Contact** nodes.

If you have many contact pairs in your model, it is a good idea to manually set the **Label** of each pair in order to simplify the identification during subsequent selections in the **Contact** nodes.

AUTOMATIC GENERATION OF CONTACT PAIRS

Contact pairs can be automatically generated during the finalization of the geometry sequence. When **Action** is set to **Form an assembly**, you can select **Create pairs**, and use **Contact pair** as **Pair type**. Boundaries which are in geometrical contact with each other will then be placed in contact pairs. All contact pairs created are automatically added to the default **Contact** node. If you do not want to use all pairs, either delete the pairs from the model or disconnect the default **Contact** node in the relevant physics interface by selecting **Disconnect pair**. For the latter alternative, add new **Contact** nodes in the physics interface, and select which of available pairs to actually use for the contact analysis.

The automatic pair generation will not know which side to use as source or destination. Based on the suggestions in Selecting Source and Destination above, you may need to switch selections using the **Swap Source and Destination** ( ) button in the **Source Boundaries** section of the contact pair settings.

SMALL SLIDING CONTACT

In some situations, the relative sliding between the contacting boundaries is small. This is, for example, often the case for shrink fit simulations, when mounting a component using prestressed bolts, or for partial decohesion of two components. The sliding distance can be considered as small if it is significantly less than the length of a mesh element edge.

In such cases, it is possible to simplify the problem by selecting the **Mapping Method** to be **Initial Configuration** in the **Contact Pair** node. With this setting, a point on the destination boundary is mapped to the same point on the source boundary throughout the entire simulation; that is, the mapping is constant. This setting will make the contact analysis run faster and be more stable.

The analysis is geometrically nonlinear also when using this option, and the contact region can still have arbitrarily large displacements and rotations.

Friction can be modeled. Even though there is no sliding in a geometrical sense, the difference in tangential displacement is computed.

You cannot mix contact pairs with the two different types of **Mapping Method** within the same **Contact** node.

General Contact Pairs

General contact pairs ( ) are only used by **General Contact** nodes.

CONTACT SELECTION

A **General Contact** node can reference any number of general contact pairs. However, when adding a **General Contact Pair**, the contact selection is set to include all objects. A single pair is then sufficient to define contact interactions between all geometric objects in the model. In most cases, this is the intended behavior.

For complex or large models there are situations where you may want to take manual control of the contact selection. A typical example is fluid–structure interaction and similar multiphysics. Here you want to exclude the fluid domains from the general contact pair to correctly identify the exterior boundaries of the solid objects.

Other cases are when you know that some parts of the model may not be of interest for contact analysis or never interact. The cost of the contact analysis can then be reduced by using a general contact pair with a manual selection for each region.

Lastly, if some boundaries cause difficulties with convergence for the contact analysis they can be omitted by using a manual control of the contact selection.

In all above cases, when using a manual contact selection, the resulting source and destination boundaries are set up from the union of the exterior boundaries of the selected domains and the selected boundaries.

The selection of a general contact pair often intersects with the selection of other pairs in the model. While this is not an issue in most cases, it may cause interactions that are either unnecessary to compute or even unphysical. To avoid this, it is possible to

exclude boundaries from other pair types in the source and destination boundaries generated for the general contact pair. By default, destination boundaries from other pair are excluded but it is also possible to exclude source boundaries from other pairs.

SMALL SLIDING CONTACT

In some situations, the relative sliding between the boundaries in contact is small. This is, for example, often the case for shrink fit simulations, when mounting a component using prestressed bolts.

The sliding distance can be considered small if it is significantly less than the length of a mesh element edge. In such a case, it is possible to simplify the problem by selecting the **Mapping Method** to be **Initial Configuration** in the **General Contact Pair** node. With this setting, a point on the destination boundary is mapped to the same point on the source boundary throughout the entire simulation; that is, the mapping is constant. This setting will make the contact analysis run faster and be more stable.

The analysis is geometrically nonlinear also when using this option, and the contact region can still have arbitrarily large displacements and rotations.

Friction can be modeled. Even though there is no sliding in a geometrical sense, the difference in tangential displacement is computed.

You cannot mix contact pairs with the two different types of **Mapping Method** within the same **General Contact** node.

Meshing for Contact Analysis

This section is only relevant for **Contact**.

Once the source and destination boundaries are chosen, you should mesh the destination finer than the source. Do not make the destination mesh just barely finer than the source because this can cause unphysical oscillations in the contact pressure. Make the element size on the destination at least two times finer than on the source. The reason is that the algorithm is asymmetric; the points on the destination side connects to the source side, and not vice versa. With a coarse mesh on the destination side, a large portion of an element (or even a whole element) on the source side could be without connection to the destination.

It is always important that the geometry is well resolved, so that a curved contact boundary actually is seen as curved rather than "faceted". The density of the mesh often needs to be finer than what would be needed to resolve stresses on a similar boundary without the contact conditions. If the normal to the contact boundary changes much from one element to the next, there is a risk that the contact analysis does not converge.

If the source boundary is rigid, there are no requirements on the mesh size of the destination boundary. In this case, you can use a significantly finer mesh on the source boundary than on the destination boundary. This is sometimes necessary in order to resolve the geometry well. On the other hand, if you have a flat rigid boundary, you only need to mesh it with a single element.

Settings for Contact Nodes

THE CONTACT AND INTERIOR CONTACT NODE

Penalty Factor

An important parameter in the settings for the Contact node is the penalty factor. When running into convergence problems, check the penalty factor settings and consider changing the current value. It is used by all contact methods, but its interpretation differs:

- In the penalty method, the penalty factor has a straightforward interpretation as the stiffness of a distributed spring connecting the two contacting boundaries. A larger value will decrease the unphysical penetration, but can be detrimental to the numerical conditioning of the stiffness matrix. On the other hand, a value that is too small can result in violation of the contact condition.
- In the augmented Lagrangian method, the penalty factor is a numerical parameter, which affects the convergence properties of the algorithm. Its interpretation is different depending on the chosen solution method. For a segregated method, the penalty factor mainly affects the rate of convergence of the outer iteration loop; a higher value typically leads to faster convergence, but can decrease the stability of the algorithm for certain problems. For the coupled method, the penalty factor mainly affects the structure of the underlying equations, and the solution is typically less sensitive to the value of the penalty factor. However, the convergence properties can be improved by tuning the penalty factor.
- In the Nitsche method, the penalty factor is a stabilization parameter which can affect the robustness of the solution, but to a lesser extent the accuracy. The suitable

value of the penalty factor depends on the chosen formulation, where the skew-symmetric and incomplete formulation are typically less sensitive than the symmetric formulation.

In the augmented Lagrangian method and the Nitsche method, the value of the penalty factor does not affect the accuracy of the final solution, like it does in the penalty method.

High values of the penalty factor can lead to an ill-conditioned stiffness matrix and convergence problems in the Newton iterations. This is often identified by that the damping factor reported by the solver becomes less than 1 for many Newton iterations, or by that the structure "jumps" into an unphysical state. Large errors returned from the linear solver are also an indication that the stiffness matrix is ill-conditioned. If this occurs, decrease the penalty factor.

The default value for the penalty factor is based on a *characteristic stiffness*. The default is an "equivalent" Young's modulus (`Eequ`) of the material on the destination side. For linear elastic isotropic materials, `Eequ` is the actual Young's modulus. For other types of materials, `Eequ` is defined by an estimate of a similar stiffness at zero strain. For materials that are user defined or in other ways nonstandard (for example, anisotropic with large differences in stiffness in different directions), `Eequ` might need to be replaced with another estimate.

For nonlinear materials in general, and for elastoplastic materials in particular, there can be a significant change in stiffness during the solution process. Choose the source and destination boundaries accordingly. You may even have to adjust the penalty factor as the solution progresses when such materials form a contact boundary.

When using the augmented Lagrangian formulation with a segregated solution method, having a well-tuned penalty factor is important for the performance of the outer contact iterations. The default value is selected as a compromise between speed and stability, but with more weight on stability. The strategy is for each new step (parametric step or time step) to start with a softened penalty factor, which is then increased over the first four iterations. The purpose is to stabilize the problem in case there are large overclosures during the first iterations. This is called *relaxation*.

In a situation where the contact is well established, using relaxation will however cost extra iterations, and it may even lead to a loss of convergence.

For this method, the penalty factor can be tuned in several ways. You have three basic choices, ranging from simple to advanced:

- With a **Preset** penalty factor, you can choose having it tuned for **Stability**, **Speed**, or **Bending**. With **Stability**, relaxation is used in every step. With **Speed**, a constant penalty factor is used, and the value that is used is equal to the final value obtained when using **Stability**. For **Bending**, a constant and low penalty factor is used in all iterations. The value corresponds to the initial value when using **Stability**. As the name implies, this option is intended for bending dominated problems where the structural stiffness is much lower than the bulk stiffness of the material.
- With **Manual tuning**, you can adjust the magnitude of the penalty factor, and also the relaxation strategy.
- With **User defined**, you can enter any expression for the penalty factor.

For details about these settings, see the documentation of the settings for the Contact node.

Some hints for selecting the penalty factor for the segregated augmented Lagrangian method:

- Use relaxation only when large changes in the contact state is expected.
- If an analysis takes a long time, check the convergence graphs. If the contact variables show a steady, but slow, convergence it may help to increase the penalty factor. Increase by a factor of $2 - 5$.
- If a model fails to converge, and the contacting parts appear to have moved far from each other, it is probable that the penalty factor is too high. You can then either decrease the total stiffness or add more relaxation.

Penalty Function

The standard choice is to use a linear ramp function by which the contact pressure is linearly related to the gap when contact is active. This is a simple and robust choice that works well for the majority of situations. An alternative is to apply a smooth transition at the initiation of contact by using a smooth ramp function. A smooth penalty function is particularly useful for structural dynamics where it can stabilize the contact state and alleviate noise in the contact pressure related to impact events, but can also be useful in other situations.

Trigger Cutback

This option is only available for **Contact**, but not for **Interior Contact**.

If, during the iterations, a contact problem comes into a state where it is far from the converged solution, it is unlikely that the solution will ever converge. In such a case, much computing time can be spent before the maximum number of iterations is reached, and the solver tries a smaller time or parameter step. To speed up this process, you can select the **Trigger cutback** checkbox when using the augmented Lagrangian method. You can then enter a logical expression that will force the solver to immediately abandon the iterations and try a smaller step when fulfilled. Such an expression can, for example, be a maximum displacement (like `solid.disp > 5[mm]`), based on what is physically realizable for the structure. The expression is evaluated in all points on the boundary, but you can also use a global value, like an average or a maximum.

Contact Surface Offset and Adjustment

For **Contact**, it is possible to assign an offset to both the source and destination boundaries. For **Interior Contact**, a singe offset value can be given.

When an offset is given, the boundary used in the computations is not the geometrical boundary, but a virtual boundary displaced by the offset value. You can use this option for several purposes:

- When analyzing problems with for example shrink fit, nominal dimensions can be used for the geometry, and the overclosure included in the gap by using a positive offset.
- When there is a small clearance between two boundaries, a negative offset can be used instead of changing the geometry.
- If geometries are such that a large overclosure exists in the initial configuration, the contact algorithms may not converge. You can then add a negative offset, which is slowly removed by letting it depend on time or on the parameter in the parametric continuation solver.
- A positive offset can be used to avoid a complete collapse of a mesh that exists between the source and destination boundaries. This is discussed in more detail in Multiphysics Contact Analysis.

When the source and destination boundaries are curved, the discretization introduced by the meshing may lead to small variations in the computed distance between the source and destination boundaries, even though the geometrical shapes of the two objects are ideal. When modeling for example a shrinkage fit, this effect can cause significant fluctuations in the contact pressure. If you select **Force zero initial gap**, the computed distance from destination to source will be adjusted by the initial gap distance detected by the contact search. Positive gap distances smaller than the tolerance Δ_{gap} are adjusted to be zero. By default, Δ_{gap} is set to `Inf`, which means that all gaps and overclosures detected are adjusted to be zero. This adjustment can be combined with an offset. The offset is applied to the adjusted gap value.

The **Force zero initial gap** option is only available for **Contact**, but not for **Interior Contact**. In the latter case, there can not be any mismatch in the mesh between the two sides

It is only the gap computation that is affected by this setting, the mesh as such is not adjusted. This type of adjustment is most useful when the sliding is small, so that the gap distance is always computed between the same points on source and destination.

You can only apply an offset to the source boundaries if they belong to the same physics interface as the destination boundaries.

Initial Value

In the augmented Lagrangian method, where the contact pressure is a dependent variable, it can be given an initial value. In force-controlled contact problems where no other stiffness than the contact prohibits the deformation, the initial contact pressure is crucial for convergence. If it is too low, the parts might pass through each other in the first iteration. If it is too high, they will never come into contact.

Discretization

When using the augmented Lagrangian method it is possible to change the type and order of the shape functions used for the contact pressure and friction force degrees of freedom. The default is linear shape functions, which ensures that there are no overconstraints in the contact interface. It is allowed to use a shape function order that is equal to or lower than the order of the displacement field. Increasing the order of the contact variables from the default can increase the accuracy of how well the contact conditions are enforced, but can impair convergence and increase the computational cost.

For a discretization other than **Linear**, the lumped solver is no longer optimal for the contact pressure update when using a segregated solution method. In such cases, a standard segregated step should be used. The default solver generation takes this into account, but if you later modify the discretization, you should update the solver sequence.

Quadrature Settings

The weak equations set up by the **Contact** node and its subnodes typically involve discontinuous functions. These originate from the contact mapping, where the source and destination meshes, in most cases, are nonconforming. The default quadrature used in the numerical integration of these integrals is equal to the order used by the displacement field. For a quadratic displacement field, this means integration order equal to four.

In most situations, the default quadrature provides sufficient accuracy of the numerical integration. However, it is sometimes necessary to increase the integration order for the contact weak equations. This can improve the stability of the contact model since it reduces numerical errors that might occur during assembly of the system matrices. Note that using an integration order that is too high can significantly increase the cost of assembling the matrices.

Jacobian Contribution

In the Advanced section of the **Contact** node, there is an option to specify the type of **Jacobian contribution** from the contact equations. The default **Automatic** option will choose a suitable setting depending on the mapping method used by the contact pair. However, if controlled manually the **Nonsymmetric** option is the preferable choice especially when the source boundaries undergo large deformations, since it is more robust. The **Symmetric** option can be attractive for large models since it preserves the symmetry of the global stiffness matrix, as long as no other features cause it to be nonsymmetric. This can decrease the solution time and memory requirements when solving the model.

THE FRICTION NODE

When adding a Friction node, you can specify a constitutive model (friction model) for the behavior of the tangential contact. This model includes conditions for switching between sticking and sliding, as well as computation of the current friction forces.

Friction Parameters

The friction model specifies the threshold for the friction force in the contact pair. If the computed (trial) friction force is above this threshold value, the contact is in a stick

state; if the (trial) friction force is above the threshold, the contact is in a slip (or sliding) state.

Two predefined friction models are available based on the classical Coulomb law, where the friction force is proportional to the contact pressure through the friction coefficient. Both Coulomb laws are additionally generalized by allowing specification of minimum (cohesion) sliding resistance and a cap that sets the maximum tangential traction.

It is also possible to define the threshold for the friction force as an arbitrary expression that may depend on any quantity in the model, for example temperature or position. The only limitation is that the expression may not implicitly depend on the current value of the friction force.

Friction Force Penalty Factor Control

This section provides similar settings as described in Penalty Factor of the **Contact** node, but the penalty factor is here used to regularize the stick constraint. However, the same considerations for how to set an appropriate value apply. For convenience, it is also possible to utilize the penalty factor set in the parent contact node.

This section is not available when the Nitsche method is used. The same penalty factor as set in the parent **Contact** node will be used for friction as well.

THE SLIP VELOCITY NODE

The Slip Velocity node facilitates a simplified form of slip friction modeling, which can be used in the case that the direction and speed of the sliding is known. The same friction models as for the **Friction** node are available. However, it is here assumed that the tangential contact is in a sliding state, and that the slip velocity is known beforehand. The latter is supplied to the feature as a user input in the local coordinate system.

Knowing the slip velocity greatly simplifies the computation of friction forces. There is no need to determine the transition between stick and slip contact, which can be difficult.

THE ADHESION NODE

When using the penalty method, you can specify that the boundaries in the contact pair should stick to each other after coming into contact by adding an Adhesion node.

The adhesive layer is conceptually without thickness, but by specifying an offset in the **Contact** node, you can to some extent include the dimensions of the adhesive layer.

The adhesive layer always has a finite stiffness. For a glue layer, this represents the true stiffness. For a more conceptual joining of two boundaries, this stiffness should be considered in the same way as the penalty stiffness in the contact formulation. The stiffness can differ between tension and compression: In compression the stiffness is always taken as the penalty stiffness, whereas you can change the tensile stiffness.

THE DECOHESION NODE

When adhesion is active, it is possible to break the bond between the source and destination boundaries by adding a Decohesion subnode.

- When used with **Contact**, **Decohesion** is a direct child node. It requires that an **Adhesion** subnode is present and active under the same parent **Contact** node.
- When used with **Interior Contact**, **Decohesion** is added as a child to an **Adhesion** node.

Decohesion defines a cohesive zone model (CZM) based on interface damage mechanics on the adhesive layer. Damage is assumed to be a scalar variable that initiates as 0 and grows to 1 during decohesion, and in principle degrades the stiffness of the adhesive layer. Since damage is a scalar, both the normal and tangential stiffness components degrade simultaneously, irrespective of whether the actual loading direction. However, the penalty stiffness of the contact condition is not affected by damage.

Two alternative CZM are available. The **Displacement-based damage** models defines damage growth as a function of a mixed mode displacement quantity. It comes with several traction separation laws that associate the onset of damage with the peak strength of the interface. For some of them, it is possible to choose between different mixed mode failure criteria. The **Energy-based damage** models define damage growth as a function of the stored undamaged elastic energy density of the interface. It also comes with several different traction separation laws. However, these are more general and define the onset of damage at an arbitrary elastic energy density. In principle, you can define the model so that damage initiates immediately during loading of the adhesive layer, that is for zero energy density. The strength of the interface is then determined by the critical energy release rate and the shape of the damage evolution function. In this way, the energy-based damage models can be viewed as a regularization of linear elastic fracture mechanics.

Decohesion is an inherently unstable process. The elastic energy in the strained adhesive layer is released during decohesion. Numerically, the decreasing branch of the traction-separation curve manifests itself as a local negative stiffness. Such problems are only possible to solve as long as the surrounding material can absorb the released

energy. The numerical stability is, furthermore, closely coupled to the physical stability of the structure. The following points can help to set up a model with decohesion and to overcome problems with convergence.

- If the structure is in a load-controlled situation, there is no possibility to continue the analysis when the peak external load has been exceeded. Physically, this corresponds to a sudden collapse of the component, or in the case of decohesion, a rapid breakage of the adhesive layer.
- If the structure is loaded by prescribed displacements, it is usually possible to continue the analysis further. It may, however, happen that the stored elastic energy in the structure is large enough to force a complete breakdown of the adhesive layer once a certain external displacement is reached. If possible, use prescribed displacements in a decohesion analysis, and evaluate the applied load from the reaction forces.
- Sometimes it is not possible to use prescribed displacements, for example if the load is distributed. You can then add a **Global Equation** to control the loading rate by some other quantity that increases monotonically. This is the same technique as the one used for postbuckling problems.
- To improve the robustness of the solver, it is sometimes beneficial to modify the settings in the **Method and Termination** section of the **Fully Coupled** or **Segregated** nodes in the solver sequence. For example, allow a larger number of iterations or try a different nonlinear method. Often, the **Constant (Newton)** method can improve the convergence of models with decohesion.
- The robustness of the solver can also be improved by modifying the parameter or time stepping algorithm. For a stationary study, you can tune the step size in the **Parametric** node, and for a time-dependent study, you can modify the time stepping of the **Time-Dependent Solver**. A good idea is often to reduce the maximum allowed step size of the solver and to allow for smaller step size than the default. Note that if the maximum step size allowed is too large, the solver might bypass the decohesion process altogether; in other words, even though a converged solution is obtained, it might be invalid.
- The solution of the unstable failure due to decohesion is, to some degree, always mesh dependent; see, for example, Ref. 1. It is therefore good practice to make sure that the mesh of the interface and in its vicinity is fine enough to allow the energy released during decohesion to properly redistribute in the structure. This can help avoid solution jumps where several mesh elements are completely damaged in a single step. Such solution jumps can be difficult for the solver to get pass, and even if it does, the solution after the jump might be invalid.

- The true unstable failure of decohesion is a dynamic event. This can be analyzed using a time-dependent solver, but the computational cost can be high. The inertial forces will then balance the released energy, which is transformed to kinetic energy.

For time-dependent studies, it is possible regularize the CZM with a viscous delay by selecting **Delayed damage** in the **Regularization** list. This option adds a delay to the release of energy, which is controlled by the **Characteristic time** τ. Using this option can help to suppress the instability of the solution when the step size or mesh size is too large. If the viscous damage is used to stabilize a rate-independent decohesion problem, the value of τ must be chosen with care. As a rule of thumb, τ should at least be one or two orders of magnitude smaller than the expected time step.

For an example showing a decohesion analysis, including how to use a global equation to control an unstable problem, see *Mixed-Mode Debonding of a Laminated Composite*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/cohesive_zone_debonding**.

THE WEAR NODE

By adding a Wear subnode to a **Contact** node, it is possible to model adhesive or abrasive wear of the material when the contacting boundaries are sliding along each other. Since wear involves solving evolution equations, the **Wear** node only adds a contribution for time-dependent studies. Moreover, wear is typically a slow process where dynamic effects are of small significance. You should, therefore, usually set **Structural transient behavior** to **Quasistatic** in the Structural Transient Behavior section of the physics interface settings.

The most general technique to model the removal of material during the wear process relies on the deformed geometry concept. When selecting the **Deformed geometry** formulation, the wear feature adds a (hidden) Deforming Domain feature that controls the material frame through an adaptive mesh smoothing. The wear, as computed in the **Wear** node is fed as a (hidden) Prescribed Normal Mesh Displacement boundary condition to the deforming domain, and thus describes the actual removal of material from the geometry. When using this formulation, you must be aware that the adaptive mesh means that state variables stored in Gauss points, for example plastic strains or creep strains, will not represent the same material points all the time. Whether or not this effect is acceptable must be judged on a case-by-case basis. Unless the amount of material that is removed is large, or gradients are strong, this is mainly an issue close to the boundary where material is removed by the wear process.

Alternatively, wear can be incorporated as an offset in the contact condition. This formulation is computationally less expensive, and is suitable as long as only small amounts of material are removed, and the wear does not change the orientation of the normal to the boundary significantly.

The slip velocity used for the wear computation can be obtained from either a Friction node or a Slip Velocity node, so one of these two nodes should be present and active under the same **Contact** parent node. For the **Generalized Archard** wear model, this is a requirement. In most cases, the orientation of the slip velocity is known a priori in a wear analysis, in which case **Slip Velocity** provides the more efficient solution.

In general, modeling wear on the destination side is slightly more accurate, since it is there that the contact pressure and slip velocity are originally computed. When modeling wear on the source side, these quantities are mapped from the destination boundary. Multiple **Wear** nodes under the same **Contact** node contributes with each other, which means that it is possible to model wear on both source and destination simultaneously. However, adding multiple wear contributions to either source or destination may give unphysical results. On the source side it is possible to also use the Rigid Material material model; this is not permitted on the destination due to general restrictions of the **Contact** node.

- Documentation of the Contact, Friction, Slip Velocity, Adhesion, Decohesion, and Wear nodes.
- Contact Analysis Theory in the *Structural Mechanics Theory* chapter.

Settings for General Contact Nodes

THE GENERAL CONTACT NODE

The General Contact node identifies the boundaries of the physics interface that are included in the contact problem. It can control some fundamental settings of its subnodes.

Quadrature Settings

The **General Contact** node can control the quadrature setting used by all its subnodes. See Quadrature Settings.

Jacobian Contribution

The **General Contact** node can control the Jacobian contribution for all its subnodes. See Jacobian Contribution.

THE CONTACT MODEL NODE

The Contact Model node is responsible for implementing the contact condition on its selected boundaries using the penalty method. This includes the definition of the contact pressure.

Penalty factor

The penalty factor is one of the most important settings for a contact analysis. See the general discussion on selecting a penalty factor for the penalty method in Penalty Factor. General contact provides many settings to control the magnitude of the penalty factor. If different controls are required for different parts of the model, add multiple **Contact Model** nodes.

The penalty factor can either be defined using estimates based on the stiffness or mass from the domain material adjacent to the contact boundary. In most cases the stiffness estimate results in a good penalty factor, but for structural dynamics problems that include soft materials, the mass estimate can result in a better penalty factor. In both cases the final value can be tuned by a user defined multiplier.

General contact can add forces on both sides of a mapping. Hence it is important to consider the material properties on both source and destination of the mapping. By default, a harmonic mean is used such that the final penalty stiffness k equals

$$k = \frac{2}{\frac{1}{k_{\text{src}}} + \frac{1}{k_{\text{dst}}}}$$

where k_{src} and k_{dst} are stiffness estimates on source and destination sides of the map, respectively. Each estimate can be either from mass of from stiffness. The harmonic mean skews the penalty stiffness toward the softer material if there is a high contrast between source and destination. This is important to avoid instabilities due to for example ill-conditioned matrices. It is also possible to set k equal to the minimum of k_{src} and k_{dst}, or to use $k = k_{\text{dst}}$. For a contact mapping between a destination and a source without physics, $k = k_{\text{dst}}$.

Penalty Function

See Penalty Function.

THE DAMPING NODE

Contact damping can be added to the contact pressure by adding a Damping (General Contact) node. In structural dynamics in general, and impact in particular, the contact state is not always stable an may shift for a given point between time steps or even

iterations. This can cause high-frequency noise in the contact pressure that propagates in to adjacent domains. Adding a damping force can help to stabilize the contact state and thus alleviate some of this noise. However, adding to much damping can have negative effect on the solution and significantly alter the dynamics properties of the contact.

THE FRICTION NODE

When adding a Friction (General Contact) node, you can specify a constitutive model (friction model) for the behavior of the tangential contact. This model includes conditions for switching between sticking and sliding, as well as computation of the current friction forces. See Friction Parameters for a discussion o the relevant settings.

A general contact problem can involve the interaction between many different materials where each interaction might require different friction parameters. For such situations, the friction parameters can defined conditionally on the contact mapping using the `src2dst` operator and relevant information about the mapped boundaries. Some examples include:

- The `dom` variable returns the entity index of the boundary where it is evaluated. Similarly, `mean(dom)` returns the entity index of the adjacent domain to the boundary where it is evaluated.
- The `concompid` variable returns a unique index for the geometric object to which the boundary belongs.
- The general contact pair adds variables to boundaries attached deformable or rigid domains as defined by a physics interface.
- Create a user defined a variable using **Variables** nodes under **Definitions** with a unique definition on different boundaries.

Any of these variables can then be used to create the conditional definition of the interaction property. Examples for the friction coefficient:

```
if(src2dst_p1(dom)==2 && dom==3 || src2dst_p1(dom)==3 && dom==2, 0.7, 0.3)
if(src2dst_p1(p1.deformable), 0.7, 0.3)
if(src2dst_p1(fricidx)==1 && fricidx==2 || src2dst_p1(fricidx)==2 &&
fricidx==1, 0.7, 0.3)
```

where in the last example `fricidx` is a user defined variable. More elaborate definitions can be set-up by using the `nif`-operator, or using functions.

THE OFFSET NODE

Adding an Offset node makes it possible to assign offset to the selected contact boundaries. The offset is considered on both source and destination of a contact

mapping that includes the selected boundaries. See further details in Contact Surface Offset and Adjustment.

The **Offset** node also allows to modify the search distance defined by a general contact pair on its boundaries. A typical where this might be useful is if the model includes coarsely meshed thin domains stacked on each other in the initial or current configuration. The contact search can for such cases sometimes find unwanted mappings even with the built-on intersection checks of the general contact pair. A remedy that often works is then to use a smaller search distance.

Quasistatic Contact Analysis

When including contact in a stationary or quasistatic simulation, make sure that the bodies are sufficiently constrained. If the bodies are not in contact in the initial configuration, and there are no constraints on them, there will be possible rigid body displacements. This will cause the solver to fail and must be avoided.

For a more detailed discussion about sufficient constraints, see Constraints under Stationary Analysis.

Sometimes, as when simulating mounting processes, the structure is not fully constrained until the contact is fully established. There are some strategies for how you can deal with this problem.

- Create the geometry or set initial values for the displacement variables so that there is a small penetration in the initial configuration.
- Use boundary conditions giving a prescribed displacement rather than a prescribed force. When possible, this is usually the best way to stabilize problem. Note that you can always obtain the force actually applied from the reaction forces.
- Add a Stabilization subnode.
- Add a temporary weak spring during the beginning of the simulation. Assuming that a parameter p, ranging from 0 to 1, is used for applying the external load, you can introduce a stabilizing spring with stiffness k_x in the x direction as

$$k_x = k(1-p)2^{-(p \cdot 10)}$$

and similarly in any other direction that needs to be restrained. It is not important whether the spring is applied to domains, boundaries, or edges, but it should not create significant local forces. The value for the stiffness k should be chosen so that

the force generated by the spring balances the external load at a sufficiently small displacement. A spring that is too weak will give an initial overclosure of the contact boundaries that is too large. In contrast, a spring that is too stiff may influence the solution too much.

Implicit Dynamic Contact Analysis

The contact formulations in COMSOL Multiphysics can be used in transient as well as quasistatic analyses. In truly dynamic problems, where inertial effects are significant, a contact formulation must conserve fundamental quantities, such as linear and angular momentum and energy, across the contact pair. An important class of dynamic problems, where these quantities must be conserved, is impact analysis.

To model dynamic contact events, two specialized contact methods are provided, the **Penalty, dynamic** and the **Augmented Lagrangian, dynamic** methods. Both are based on a viscous formulation that constrains the gap rate to be zero, ensuring that the normal contact is dissipative and does not introduce any spurious energy contribution to the system. Since the methods are dissipative, they are mainly intended for short duration events, such as soft impact between two bodies. For prolonged interaction between two bodies, energy dissipation can become significant, and overclosures can become large, since the gap rate is only approximately zero. Both the dissipation and the accuracy are controlled by a penalty factor that for these two methods conceptually represents a dashpot, rather than a spring. It therefore has a characteristic time user input that sets its magnitude. As a rule-of-thumb, it should be of the order of the contact event duration, but the best choice must be decided on a case-by-case basis.

The **Penalty, dynamic** method also provides the possibility to combine the stiffness and viscous based penalization of the normal contact. For impact analysis, it is often best to use only the viscous formulation by setting the stiffness **Penalty factor control** to **Viscous only**.

When modeling dynamic contact, the primary focus is usually on the kinematics between the contacting bodies. If you use the default adaptive time-stepping algorithm, the solver will also attempt to resolve wave propagation in the domains adjacent to the contact pairs. This often leads to unnecessarily small time steps, significantly increasing computation time.

To avoid this, you can switch to a manual time-stepping algorithm in the **Time-Dependent Solver** settings. Choose time steps that are small enough to accurately capture the contact event, since too large steps can introduce spurious energy contributions and cause the simulation to become unstable or diverge.

Implicit time-dependent solvers in COMSOL Multiphysics introduce numerical damping to stabilize time stepping. While this stabilization is often necessary, excessive numerical damping can remove important information from the simulation. For this reason, the BDF solver is not recommended for dynamic contact problems. Instead, when inertial terms are included, the preferred choice is the generalized-α solver.

Regardless of the solver method or settings used, it is good practice to perform an *a posteriori* check of momentum and energy conservation to ensure that the solution is acceptable.

For general contact, adding a **Damping** subnode to **Contact Model** is equivalent to selecting Penalty, dynamic method.

For examples of dynamic contact problems, see

- *Transient Rolling Contact*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/transient_rolling_contact**.
- *Impact Between Two Soft Rings*: Application Library path **Structural_Mechanics_Module/Verification_Examples/ring_impact**.

In both examples, an energy balance check is included. The latter model also verifies the conservation of momentum.

Explicit Dynamic Contact Analysis

An explicit dynamics analysis is particularity well-suited for solving complex and large-scale contact problems. This applies not only to truly dynamic phenomena such as impact but also to slower processes like metal forming. The advantages stem from some of the inherent characteristics of explicit time-stepping, including:

- Small time steps, which improve the time resolution of the contact state
- Direct evaluation of the contact state from known information, eliminating the need to iteratively find the contact state for each time step
- No nonlinear iterations required to establish equilibrium, and therefore no need to assemble or factorize a stiffness matrix

The last two items ares often the main sources of poor convergence in contact problems when using implicit methods, making explicit dynamics a compelling alternative.

In explicit dynamics, contact conditions are mainly implemented using the The Penalty Method, where an additional stiffness is introduced between the contacting boundaries. The trade-off between accuracy and stability when selecting the penalty factor becomes perhaps more apparent when using explicit time-stepping since stiffness added by contact forces also impact the stable time step. This effect is not automatically accounted for, but the default penalty factors definitions are chosen not to impact the stability of the solver. If needed, the contact stiffness can be adjusted by manually tuning the penalty factor multiplier.

The option in **General Contact** to estimate the penalty stiffness form the mass scales the penalty factor based on the current stable time step of the used by the solver.

The default settings for contact are in most case sufficient to produce stable results for the contact analysis, however, the accuracy and stability of the contact analysis are also affected by other parameters such as the time step and the mesh. If the contact part of a model appears to be the cause of inaccurate results or instabilities, consider the following:

- A simple remedy if instabilities are observed is to reduce the penalty factor, but be aware that a too small penalty factor can cause unphysical results.
- If excessive penetration or even loss of contact occurs, consider using a higher penalty factor. However, do so carefully as a too high value will cause instabilities.
- Evaluating changes to contact state and forces at smaller intervals often improves the accuracy and stability, but comes at a cost in solution time. Reducing the time step is most readily done by the **Time step safety factor** available for the Explicit Dynamics study step. The safety factor can be a function of time which allows using a stricter time step for selected intervals of the time history.
- Refining the mesh in problematic regions can help to smooth the contact forces as they are distributed over more evaluation points. An improved mesh refinement also improves the resolution of possibly problematic features of the geometry such as fillets. Keep in mind that a refined mesh can cause a reduction in the stable time step.

- An alternative to refining the mesh in problematic contact regions is to increase the quadrature order.
- Adding contact damping can help stabilize the contact state and remove high frequency oscillations from the contact forces. However, too much damping can quickly lead unrealistic contact forces and a significantly altered dynamics behavior. Monitor the energy dissipation by contact damping to verify that effects on the model response is acceptable.

Add contact damping by

- Selecting the Penalty, dynamic method in a **Contact** node.
- Adding a **Damping** subnode to a **Contact Model** node for general contact.

See also Performing an Explicit Dynamics Analysis.

Multiphysics Contact Analysis

This section is only relevant for **Contact**.

Two different classes of multiphysics contact problems will be described in this section.

FLUXES THROUGH CONTACT SURFACES

In some contact problems, there is some kind of flux from one domain to another through the contact zone. This can for example be a heat flux, an electric current, or moisture transport. The common property here is that the other physics fields than the displacements are present in domains where the solid mechanics problem is solved. Typically, there will be a more or less perfect insulation as long as there is no contact, but as soon as contact is established there will be a flux through the contact area.

This class of problems often exhibit a high degree of nonlinearity, which may lead to convergence problems in the nonlinear solver. As an example, consider heat transfer through the contact area, where initially only a small spot is in contact. The solution for the temperature field is then extremely sensitive to the size of the contact area. If,

at the same time, the solid deforms due to thermal expansion, there may be large changes in the contact area between iterations,

It is important to resolve the size of the contact area accurately, that is, to use a very fine mesh in the contact area when modeling fully coupled multiphysics problems.

If the contact area is larger, a fine mesh is not required because then the temperature solution is not that sensitive to the size. If possible, start with an initial configuration where the contact area is not very small.

You can use the contact variables (gap and contact pressure) in expressions for quantities in other physics interfaces. As an example, a thermal resistance in the contact region can depend on the contact pressure.

In many cases, the penalty method is preferred in multiphysics contact problems because of its better stability and less restrictive requirements on solver selections. If the contact conditions depend strongly on the contact pressure, use the augmented Lagrangian method because if its higher accuracy.

FIELDS EXIST IN THE GAP

In this class of problems, a field exists between the domains controlled by solid mechanics.

This is the case in, for example, Fluid–Structure interaction (FSI) problems. Here, the equations in the fluid are solved on a domain with a moving mesh, so that the shape of the fluid domain is controlled by the displacements of the solid. Another case of the same type is when there is an electric field in an air gap.

If contact is established, the mesh in the original gap between the source and destination boundaries will collapse. This must be avoided. The remedy is to add an offset in the contact settings to either the source boundary, the destination boundary, or both. If you do this, contact forces will be transmitted without the geometrical gap being fully closed.

If, for example, this technique is used when modeling a valve, there will still be some small flux even though the valve is closed, since there is a geometrical gap with the width of the artificial offset. By choosing a suitably small offset, you can however make that flux negligible.

INTERFERENCE FIT

Interference fits can be analyzed using contact modeling. This is necessary if you are interested in checking the connection with respect to, for example, slipping or local stresses.

There are two possible approaches for modeling interference fits, both equally valid:

- The actual geometry of the two parts before assembly is modeled. In this case, there will be some overlap between the two domains.
- A nominal geometry in which the contacting boundaries have the same location is created. In this case, the overlap is described as part of the contact modeling.

An imported CAD geometry can use either of these approaches, depending of the strategy used during the geometry creation. Often, the geometrical parts are modeled as nominal, and instead equipped by tolerance information that describe the amount of interference.

True Geometry

With a true geometry, you can often immediately solve the contact problem. Sometimes convergence problems may, however, appear, in particular if the material model is nonlinear. The cause is often that the initial overlap is too far from the final solution.

To deal with such a problem, add an offset to the contact boundaries. The offset should be defined by a parameter, so that the boundaries of the two domains are slightly in contact in the initial state. Now, the offset can be reduced to zero step-by-step, using an auxiliary sweep in the solver.

If you combine a true overlap with **Force zero initial gap**, the effect is that the overlap effectively is removed. In that case, you must re-introduce the overlap using an offset. This is a completely valid approach, and will often provide a more uniform distribution of the contact pressure.

Nominal Geometry

When working with a nominal geometry, you always need to add an offset to the contact boundaries. The offset equals the size of the interference. If needed for convergence reasons, ramp up the offset using an auxiliary sweep in the solver.

Interference Fit Connection in a Mountain Bike Fork: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/mountain_bike_fork**

SELF-CONTACT

To model self-contact using the **Contact** node, include the same boundaries in both the *source* and *destination* selections of the **Contact Pair** definition. This will cause the boundaries to act as both source and destination in the contact search and mapping. For mechanical contact, this results in a unbiased (or symmetric) contact formulation, as the contact conditions are formulated on both sides of the contact pair. Note that a source is not allowed to partially intersect the destination when used for mechanical contact.

Self-contact is automatically detected when using the **General Contact** node.

This technique to model self-contact means that some of the considerations discussed in this chapter regarding contact modeling do not apply. For example, instead of the recommendations in Meshing for Contact Analysis, it is recommended to use a uniform mesh element size along the contacting boundaries. Self-contact is a case where it might be necessary to increase the quadrature order used in the weak equations, see Quadrature Settings.

For examples of self-contact problems, see

- *Self-Contact of a Loaded Spring*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/loaded_spring_contact**.
- *Contact Analysis of a Rubber Boot Seal*: Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/rubber_boot_seal**.

Fallback Conditions to Contact Regions

Certain boundary conditions may act as fallback conditions to a contact region, or any pair region, if their selection overlaps with the source and destination boundaries. The Layered Shell interface does not support any fallback conditions. When a boundary condition is considered as a fallback, the condition is only active for regions that are not in contact.

In the Solid Mechanics; Solid Mechanics, Explicit Dynamics; and Multibody Dynamics interface, boundary conditions recognized as fallback conditions if their selection intersects any source and destination boundary include:

- Boundary Load
- Spring Foundation
- Added Mass
- Free

In the Shell and Membrane interfaces, following boundary conditions recognized as fallback conditions include:

- Face Load
- Spring Foundation
- Added Mass

To enable the fallback condition in these nodes, change the **Allowed region** to **Fallback and nonpair regions** in the Applicable Pair Regions section of settings window of the node. To display the Applicable Pair Regions section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The most common case when a fallback condition is used is when there is a pressure load acting on the part of the boundary that is currently not in contact. In this case you would add a **Boundary Load** or **Face Load** with a selection that intersects with the source and destination boundaries.

Solver Settings for Contact Analysis

This section provides useful tips for tuning solver settings for contact analysis in general, and tips specific to different methods.

GENERAL SETTINGS

The following solver settings can help to successfully perform contact simulations in general:

- In a contact analysis, you almost invariably use an incremental approach. It is possible to solve a problem without friction in a single stationary load step, but such an approach will often fail to converge. In a stationary analysis, you should instead use the parametric continuation solver, and gradually increase the load or displacement. Enable it by selecting **Auxiliary sweep** under **Study Extensions** in the settings for the Stationary solver.
- Use a direct solver, rather than an iterative solver, as linear equation solver if the problem size allows it. Direct solvers are less sensitive and can provide better convergence.
- As a default, the *double dogleg* nonlinear solver is selected when a stationary study is generated and contact nodes are present in the model. For the majority of contact problems this solver has more stable convergence properties than the Newton solver, which is the default solver for most other problems. Using otherwise similar settings, the double dogleg solver tends to be somewhat slower than the Newton solver on problems where both solvers converge. It is, however, often possible to take larger parameter steps when using the double dogleg solver. For some problems, the Newton solver can still be the better choice, so if you experience problems using the default settings, try to switch solvers.
- For contact problems, it is often necessary to let the parametric solver use a defensive strategy when going from one parameter step to the next. This can be controlled by setting the value of **Predictor** in the **Parametric**. By default, the parametric solver will do so by setting the predictor to **Constant** when contact is present. However, it can sometimes be more efficient to use a more aggressive strategy by setting it to **Linear**.
- Always solve contact problems that involve friction or decohesion incrementally, using a parametric or time-dependent solver. The evolution of the friction forces is history dependent. For contact problems without friction, an incremental strategy is not necessary but often a good choice.
- If the model includes friction, try solving the problem without friction first if possible. When the study runs absent friction, enable friction again.
- The convergence of many contact problems can be improved by modifying the parameter or time step algorithm. For a stationary study, you can tune the step size

in the **Parametric** node, and for a time-dependent study, you can modify the time stepping in the **Time-Dependent Solver**.

- For models that include decohesion, see the suggestions under The Decohesion Node.

THE PENALTY METHOD

Since the penalty method only adds a weak contribution to the physics, there are no special solver requirements, apart from the suggestions that apply to contact analysis in general. If the simulation shows poor convergence, or even diverges, it is for the penalty method often related to the choice of the penalty factor. A penalty factor that is too large may cause ill conditioning, which can manifest itself either in poor convergence of the Newton solver or large errors in the linear solver. If this is encountered, consider using a smaller penalty factor. However, a penalty factor that is too small will deteriorate the accuracy of the simulation, and contact might even be lost if the overclosure becomes too large.

THE SEGREGATED AUGMENTED LAGRANGIAN METHOD

The segregated solution method for the augmented Lagrangian formulation introduces extra contact degrees of freedom for the contact pressure and the friction forces. It also relies on a special solver sequence to obtain the correct solution. Some general considerations when using this method are:

- The convergence check relies on the scaling of the degrees of freedom. Since contact pressures and friction forces often are zero over parts of the simulation you should not rely on automatic scaling. When the solver sequence is first created, both contact pressure and friction forces are given manual scalings relevant for typical metal-to-metal contact. You should in most cases change these to values appropriate for your application. Variable scaling is accessed under **Dependent Variables** in the solver sequence. Set the scale for each variable to a value that is representative of the expected result. Values that are too large may result in solutions that converge too early, while values that are too small may lead to an excessive number of iterations.
- The default solver sequence generates one lumped step in the segregated solver for each **Contact** node. This split of variables into different lumped steps does not influence the solution as such; you can equally well group the contact variables in a single lumped step. Each lumped step will however generate an individual curve in the convergence plot, making it easier to pinpoint the source of possible convergence problems. You can also increase the granularity even more by changing **Solver log** to **Detailed** in the **Advanced** node in the solver sequence. This will give a separate convergence curve for each dependent variable.

The default solver generates a solver sequence that is stable and gives and accurate solution for the majority of contact problems. However, if convergence is difficult other settings can be tested:

- The **Segregated** solver is generated with a termination technique set to **Tolerance**. This setting ensures the convergence of the contact degrees of freedom, that is, that their value only changes within the specified tolerance during the latest segregated iteration. If, however, convergence of the contact DOFs is of less importance, you can accept the solution after *n* segregated iterations instead. To do this, set the termination technique to **Iterations or tolerance** and specify the maximum number of segregated iterations. The solver will then continue to the next step if the tolerance criteria is fulfilled, or if the maximum number of iterations is reached. Note that the solution of the segregated groups is still converged in each segregated iteration.
- For efficiency purposes, the nonlinear solver in the **Segregated Step** that includes the displacement field is by default set to accept the solution after seven iterations, regardless of convergence or not. If you notice in the solver log that the solution is far from convergence after these seven iterations it can be necessary to change this setting. Updating the contact DOFs with a nonconverged solution can cause the overall problem to diverge. By changing the termination technique to **Tolerance**, the segregated solver will instead do a cutback if such a situation is encountered.

THE COUPLED AUGMENTED LAGRANGIAN METHOD

The coupled solution method for the augmented Lagrangian formulation introduces special contact degrees of freedom for the contact pressure and the friction forces. Otherwise it places no special restriction on the solver sequence. Some general considerations when using this method are:

- The convergence check relies on the scaling of the degrees of freedom, but since contact pressures and friction forces often are zero over parts of the simulation, you should not rely on automatic scaling. When the solver sequence is first created, both contact pressure and friction forces are given a manual scaling typical for metal-to-metal contact. You should in most cases change this to values appropriate for your application. The variable scaling is accessed under **Dependent Variables** in the solver sequence.
- Since the solution to the augmented Lagrangian can be nonsmooth, the default *double dogleg* nonlinear solver in stationary studies is sometimes too conservative.

The convergence can in such cases often be improved by using a Newton solver, for example, the **Constant (Newton)** with a full Jacobian update.

- Although the penalty factor does not affect the accuracy of the solution, it can have significant influence on the convergence properties of the model.

THE NITSCHE METHOD

Since the Nitsche method only adds a weak contribution to the physics, there are no special solver requirements, apart from the suggestions that apply to contact analysis in general. If the simulation shows poor convergence, or even diverges, it is often related to the choice of the penalty factor. A value that is too small may cause the solution to be unstable. On the other hand, just as for the penalty method, a value that is too large may cause ill conditioning.

Monitoring the Solution

This section is only relevant for **Contact**.

It is often useful to monitor the solution during a contact analysis. This can be done in different ways.

Using the **Results while solving** functionality in the study step is a good practice. You can either use a stress plot, or a plot of the contact pressure. In most cases, the scale of a deformed plot should be set to 1 when monitoring contact problems. If you select **Results while solving** in the **Segregated** node, the plot is updated after each iteration, thus allowing you to monitor the convergence in detail.

For each contact pair, two global variables that can be used in probe plots are available. These are the maximum contact pressure (`<phys>.Tnmax_<pair>`) and the minimum gap distance (`<phys>.gapmin_<pair>`).

Looking at the convergence plot will give valuable information about the convergence properties. There will, as a default, be one graph per **Contact** node in the Model Tree, which will help you pinpoint the source of a convergence problem. You can also increase the granularity even more by changing **Solver log** to **Detailed** in the **Advanced** node in the solver sequence. This will give a separate convergence curve for each dependent variable.

You can also select to include information about the contact state in the solver log. To do that, select the **Add contact status to solver log** checkbox in the **Advanced** section of the settings for the Contact node. For each contact pair, messages like

```
69 points of 120 are now in contact.
33 points started to stick. 72 points are now sticking.
12 points started to slide. 47 points are now sliding.
```

will be generated for each time or parameter step. Only changes are reported.

Dependent Variables in Contact Analysis

The **Contact** and its subnodes will generate a number of degrees of freedom depending on the settings and study type. You will see these degrees of freedom appear under **Dependent Variables** in the solver sequence. There are two types of extra variables created:

- Variables changed until convergence is reached during the iterations. These variables appear in the **Lumped Step** nodes in the **Segregated** solver or in the **Fully Coupled** node.
- Variables used to store the state, once the iterations have converged for a certain time or parameter step, so called internal degrees of freedom. Such state variables are not immediately visible in the solver sequence, but it you select the **Displacement Field** node under **Dependent Variables** in the solver sequence, you will see them listed as 'Internal variables'.

If you change settings in the **Contact** or **Friction** nodes after the solver sequence has been generated, dependent variables may be added or removed. The second case is never a problem, but when new dependent variables are created, they are not automatically added to the groups in the segregated solver. You may then encounter the error message "`Segregated solver steps do not involve all components.`" You can then either regenerate the solver sequence, or manually insert the missing variables into the **Lumped Step** node.

In Table 2-13 the dependent variables that can be created by the **Contact** and its subnodes are summarized. To shorten the variable names, the full scope has been removed. As an example, the contact pressure variable for pair `p1` in component `comp1`,

generated in the Solid Mechanics interface `solid`, will have the full name similar to `comp1.solid.Tn_p1`. In the table, it is shown as `Tn`.

TABLE 2-13: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS.

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
Tn	Contact pressure	The contact pressure in the normal direction	Contact: **Augmented Lagrangian** used	Lumped Step
Tt	Friction force	The friction force vector	Friction: **Augmented Lagrangian** used	Lumped Step
gap_old	Previous gap distance	Physical gap variable from the last converged solution,	Contact: **Penalty, dynamic** or **Augmented Lagrangian, dynamic** used	Internal state variable
cm_old	Previous mapped source coordinates	The location on the source where this point was located at last converged solution	Friction node present	Internal state variable
sliptot_old	Previous accumulated slip	Total slip in this point at last converged solution	Friction or Slip velocity: **Store accumulated slip** selected	Internal state variable
contact_old	Contact variable in previous step	Nonzero if the point was in contact at last converged solution within the friction detection tolerance	Friction node present	Internal state variable
isContact_old	Contact status variable in previous step	Nonzero if the point was in contact at last converged solution	Contact: **Add contact status to solver log** selected	Internal state variable

TABLE 2-13: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS.

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
isSliding_old	Sliding friction status variable previous step	Nonzero if the point was in a sliding state at last converged solution	Contact: **Add contact status to solver log** selected. Friction node present.	Internal state variable
isSticking_old	Sticking friction status variable previous step	Nonzero if the point was in a sticking state at last converged solution	Contact: **Add contact status to solver log** selected. Friction node present.	Internal state variable
Wfric	Frictional dissipation density	Energy dissipated by friction	Friction present and **Compute frictional dissipation** selected.	Segregated step
Wfric_old	Previous frictional dissipation density	Energy dissipated by friction at the last converged solution	Slip velocity present and **Compute frictional dissipation** selected.	Internal state variable
activation_old	Adhesion condition at previous step	Nonzero if the adhesion condition was fulfilled at the last converged solution	Adhesion node present	Internal state variable
cma_old	Previous adhesive mapped source coordinates	Position on the source where this point was located when adhesion was detected.	Adhesion node present	Internal state variable
um_max_old	Maximum displacement jump at previous step	Maximum norm of displacement jump vector reached until previous step.	Decohesion node present and **Displacement-based damage** selected	Internal state variable

TABLE 2-13: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS.

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
Ydm_max_old	Maximum damage energy at previous step	Maximum value of the stored undamaged elastic energy at previous step	Decohesion node present and **Energy-based damage** selected	Internal state variable
vdmg_old	Viscous damage at previous step	Value of the viscous damage variable at the previous step	Decohesion node present and **Delayed damage** selected	Internal state variable
Wdmg_old Wsd_old dmg_old		These variables are related to the computation of the energy dissipated by damage	Decohesion node present and **Compute damage dissipation energy** selected.	Internal state variables

Important Contact Variables

In this section you will find a summary of variables created in the contact node that can be useful in postprocessing. In Table 2-14 important field variables are listed. For the Shell, Layered Shell and Membrane interfaces, these are appended with the suffix '`_top`' or '`_bot`', depending on which side of the boundary that is in contact. All variables are also defined per contact node, for example `<phys>.<contact_tag>.Tn` is the contact pressure from the contact node with the tag `<contact_tag>`. These variables can be useful if the model includes multiple contact nodes. Table 2-15 presents global variables related to contact that can be useful for postprocessing.

TABLE 2-14: IMPORTANT CONTACT VARIABLES.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
`<phys>.Tn`	Contact pressure	Contact	Vector in global coordinates also available
`<phys>.gap`	Gap distance including offsets	Contact	
`<phys>.Ttnorm`	Friction force norm	Friction or Slip Velocity	
`<phys>.Tt`	Friction force	Friction or Slip Velocity	Vector

TABLE 2-14: IMPORTANT CONTACT VARIABLES.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
<phys>.qfric	Friction dissipation rate	Friction or Slip Velocity	
<phys>.fs	Adhesive stress	Adhesion	Vector
<phys>.us	Displacement jump	Adhesion	Vector
<phys>.adhesion	Adhesion condition	Adhesion	
<phys>.bdmg	Damage	Decohesion	
<phys>.um_max	Maximum displacement jump	Decohesion	If displacement-based damage
<phys>.Ydm_max	Maximum damage energy	Decohesion	If energy-based damage
<phys>.Wcnt	Contact energy density	Contact	
<phys>.Wfric	Frictional dissipation density	Friction or Slip Velocity	
<phys>.Wadhe	Adhesive elastic energy	Adhesion	
<phys>.Wdbmg	Damage dissipation energy density	Decohesion	

TABLE 2-15: GLOBAL CONTACT VARIABLES.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
<phys>.gapmin_<pair_tag>	Minimum gap distance, contact pair <pair_tag>	Contact	
<phys>.Tnmax_<pair_tag>	Maximum contact pressure, contact pair <pair_tag>	Contact	
<phys>.T_tot	Total contact force	Contact	Vector
<phys>.Tn_tot	Total contact force, pressure contribution	Contact	Vector
<phys>.Tt_tot	Total contact force, friction contribution	Friction or Slip Velocity	Vector
<phys>.Wcnt_tot	Total contact energy	Contact	Integration over all contact boundaries

TABLE 2-15: GLOBAL CONTACT VARIABLES.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
<phys> .Wfric_tot	Total frictional dissipation	Friction or Slip Velocity	Integration over all contact boundaries
<phys> .Wadhe_tot	Total adhesive elastic energy	Adhesion	Integration over all contact boundaries
<phys>.Wdmg_tot	Total damage dissipation energy	Decohesion	Integration over all contact boundaries

References for Contact Modeling

1. J.L. Chaboche, F. Feyel, and Y. Monerie, "Interface debonding models: a viscous regularization with limited rate dependency," *International Journal of Solids and Structures*, vol. 38, pp. 3127–3160, 2001.

Activating and Deactivating Material

In some applications, you may want to activate or deactivate material during an analysis. An example is when you want to model the addition of material during processes such as additive manufacturing or welding.

The information in this section is applicable if your license includes the Structural Mechanics Module or the MEMS Module.

The Activation subnode can be used for this purpose. You enter an activation expression to determine whether material is active or not. When this Boolean expression is satisfied, the material is activated. Rather than truly adding or removing material, the **Activation** subnode alters the stiffness and density of the material to emulate this.

It is typically required that material is activated in a state of zero stress. Therefore, **Activation** activates material in a stress-free state by removing all elastic strains present at the point of activation.

ACTIVATION

Add an **Activation** subnode when you want to activate or deactivate one or several domains selected in a **Linear Elastic Material** node. The **Activation expression** field is used to define when material should be activated, and the **Activation scale factor** is used to reduce the elastic stiffness and density of the material which is not active.

In Figure 2-27 a case is shown where the material in domains 1, 2, and 3 is to be activated when an auxiliary sweep parameter `para` exceeds the value 1.5. The activation scale factor has a default value of 10^{-5}.

The activation condition can be any type of expression or function. A common case is that it is a function of the temperature. The activation expression is evaluated in each Gauss point. This means that an element can be partially activated. If you want to force whole elements to be activated, you can for example put the activation expression inside the `centroid()` operator.

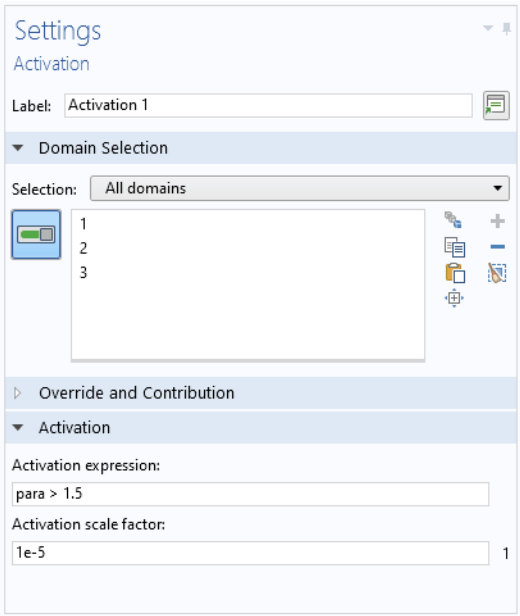


Figure 2-27: Settings for the Activation subnode.

VARIABLES

Two useful variables are created when you add the **Activation** subnode. The variable `isactive` is set to '1' when the activation condition is satisfied, and it is '0' otherwise. The variable `wasactive` is used to record if the material has been active at any previous step in the analysis. This variable can be used to "lock" the state of activation, once it has been reached. Suppose that you want the material in the previous example to remain active even if `para` later becomes less than 1.5. The activation expression for an interface with the name `solid` could then be expressed as

```
(para > 1.5) || solid.wasactive
```

Similarly, the variable `wasinactive` is used to record if the material has been inactive at any previous step in the analysis.

Domains that are not selected in any **Activation** subnode always have the `isactive` variable set to '1'.

TABLE 2-16: VARIABLES DEFINED BY THE ACTIVATION SUBNODE.

VARIABLE	DESCRIPTION
`<physics>.isactive`	Current state of the material (1 or 0)
`<physics>.wasactive`	Variable set to 1 if `isactive` has been 1 previously
`<physics>.wasinactive`	Variable set to 1 if `isactive` has been 0 previously

RESULTS

If you have performed an analysis in which only part of the material is active, it is useful to apply a **Filter** and only display the regions that actually are active; see Figure 2-28. When an **Activation** subnode is added in a Solid Mechanics or Membrane interface, such a **Filter** node is automatically added to the default stress plots.

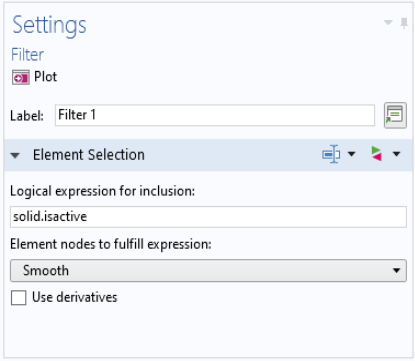


Figure 2-28: Filter settings to only display active material.

Springs and Dampers

The Spring Foundation and Thin Elastic Layer nodes supply elastic and damping boundary conditions for domains, boundaries, edges, and points.

The features are completely analogous, with the difference that a **Spring Foundation** connects the structural part on which it is acting to a fixed "ground," while the **Thin Elastic Layer** acts between two parts, either on an interior boundary or on a pair.

A **Spring Foundation** is most commonly used for simulating boundary conditions with a certain flexibility, such as the soil surrounding a construction. Another important use is for stabilizing parts that would otherwise have a rigid-body singularity. In this case a **Spring Foundation** acting on the entire domain is useful because it avoids the introduction of local forces.

A **Thin Elastic Layer** used as a pair condition can simulate thin layers with material properties that differ significantly from the surrounding domains. Common applications are gaskets and adhesives.

When a **Thin Elastic Layer** is applied on an interior boundary, it usually models a local flexibility, such as a fracture zone in a geological model.

The following types of data are defined by these nodes:

- Spring Data
- Loss Factor Damping
- Viscous Damping

SPRING DATA

The elastic properties can be defined either by a spring stiffness or by a force as function of displacement. The force as a function of displacement can be more convenient for nonlinear springs. Each spring node has three displacement variables defined, which can be used to describe the deformation dependency. These variables are named `<interface_name>.uspring1_<tag>`, `<interface_name>.uspring2_<tag>`, and `<interface_name>.uspring3_<tag>` for the three directions given by the local coordinate system. In the variable names, `<tag>` represents the tag of the feature defining the variable. The tag could, for example, be `spf1` or `tel1` for a Spring Foundation or a Thin Elastic Layer, respectively. These variables measure the relative extension of the spring after subtraction of any predeformation.

In addition to explicitly supplying a spring stiffness, you can choose to enter elastic material data and the layer thickness. The spring stiffness is then computed internally, based on an assumption of plane strain conditions.

LOSS FACTOR DAMPING

The loss factor damping adds a loss factor to the spring data above, so that the total force exerted by the spring with loss is

$$\mathbf{f}_{sl} = (1 + i\eta)\mathbf{f}_s$$

where $\mathbf{f}_s$ is the elastic spring force, and η is the loss factor.

Loss factor damping is only applicable for eigenfrequency and frequency-domain analysis. In a time-dependent analysis the loss factor is ignored.

VISCOUS DAMPING

It is also possible to add viscous damping to the Spring Foundation and Thin Elastic Layer features. The viscous damping adds a force proportional to the velocity (or in the case of Thin Elastic Layer: the relative velocity between the two boundaries). The viscosity constant of the feature can be made dependent on the velocity by using the variables named `<interface_name>.vdamper1_<tag>`, `<interface_name>.vdamper2_<tag>`, and `<interface_name>.vdamper3_<tag>`, which contain the velocities in the three local directions.

Bracket — Spring Foundation Analysis: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_spring**

Spring Foundation and Thin Elastic Layer

Defining Multiphysics Models

The following modeling tips are about how to define multiphysics models. A good place to start reading is in Building a COMSOL Multiphysics Model in the *COMSOL Multiphysics Reference Manual*.

In this section:

- Thermal–Structure Interaction
- Acoustic–Structure Interaction
- Thermal–Electric–Structural Interaction

Thermal–Structure Interaction

The Thermal Stress, Solid Interface included with this module has a predefined one-way approach for thermal–structure interaction (thermal stress), which combines a Solid Mechanics interface with a Heat Transfer interface from the Heat Transfer Module or COMSOL Multiphysics.

There are also similar multiphysics interfaces available for thin structures, as described in The Thermal Stress, Shell Interface, The Thermal Stress, Membrane Interface, and The Thermal Stress, Layered Shell Interface. The latter requires the Composite Materials Module.

By default, COMSOL Multiphysics takes advantage of the one-way dependence and solves the problem sequentially using the segregated solver. The solution for the temperature is separated from the stress–strain analysis, which then uses the computed temperature field from the heat transfer equation.

Using a single iteration in the segregated solver does not produce a correct result if there are thermal properties that depend on the displacements. Examples are when a heat source causes mechanical losses (damping) in the material or when thermal contact is present.

Acoustic–Structure Interaction

When the Structural Mechanics Module is used together with an acoustics physics interface from the Acoustics Module, it is possible to model a wide range of Acoustic–Structure interaction problems.

There are several physics interfaces available that are documented and described in the Acoustic–Structure Interaction Interfaces chapter in the *Acoustic Module User's Guide*

- The Acoustic–Solid Interaction, Frequency Domain Interface
- The Acoustic–Solid Interaction, Transient Interface
- The Acoustic–Piezoelectric Interaction, Frequency Domain Interface
- The Acoustic–Piezoelectric Interaction, Transient Interface
- The Solid Mechanics (Elastic Waves) Interface
- The Elastic Waves, Time Explicit Interface
- The Poroelastic Waves Interface
- The Acoustic–Solid–Poroelastic Waves Interaction Interface
- The Acoustic–Poroelastic Waves Interaction Interface
- The Acoustic–Shell Interaction, Frequency Domain Interface
- The Acoustic–Shell Interaction, Transient Interface
- The Thermoviscous Acoustic–Solid Interaction, Frequency Domain Interface
- The Thermoviscous Acoustic–Shell Interaction, Frequency Domain Interface

Thermal–Electric–Structural Interaction

The Joule Heating and Thermal Expansion Interface enables thermal–electric–structural interaction. This is a combination of three physics interfaces: Solid Mechanics, Heat Transfer in Solids, and Electric Currents.

The thermal–electric coupling is bidirectional, with Joule heating and temperature-dependent electrical properties, while the temperature coupling to the Solid Mechanics interface is unidirectional.

By default, COMSOL Multiphysics takes advantage of the one-way dependence and solves the problem sequentially using the segregated solver. Temperature and electric potential are solved using a coupled approach and then the stress–strain analysis uses the computed temperature field from the heat transfer equation.

Using a single iteration does not produce a correct result if there are thermal properties or electrical that depend on the displacements, making the thermal-structure part into a bidirectional coupling.

Thermally Coupled Problems

A wide class of structural mechanics problems are related to effects of variations in temperature. In this section various such effects are discussed.

Temperatures can either be computed using another physics interface, usually Heat Transfer in Solids, or directly be prescribed in the input for the various physics nodes.

For information about the predefined coupling between the Solid Mechanics and Heat Transfer in Solids interfaces, see The Thermal Stress, Solid Interface.

In this section:

- Temperature-Dependent Material Data
- Thermal Expansion
- Constraints and Thermal Expansion
- Thermoelastic Damping

Temperature-Dependent Material Data

Many material properties, such as Young's modulus, coefficient of thermal expansion, and yield stress, can have a significant dependence on temperature. In many cases, materials supplied in the material libraries and databases have such dependencies incorporated.

Materials in the *COMSOL Multiphysics Reference Manual*

If a material property under the Materials branch has a temperature dependence, you have to input the temperature to be used in the **Model Input** section in the settings window for the node in the physics interface that references the property. It is possible that not all aspects of a material are defined in the same node in the Model Builder tree. For example, if a problem is run with thermal expansion and plasticity, then:

- Young's modulus, Poisson's ratio, and mass density are given in the Linear Elastic Material node.

- Yield stress and hardening function are given in the Plasticity node.
- Coefficient of thermal expansion is given in the Thermal Expansion (for Materials) subnode.

The **Plasticity** node is available as a subnode to the **Linear Elastic Material** node when you have either the Nonlinear Structural Materials Module or the Geomechanics Module.

For each of these nodes, there is a **Model Input** section in the Settings window. Some of these sections may be empty if none of the properties given in that node have a temperature dependence. In general, you have to supply the temperature in all the **Model Inputs** sections.

As a default, the value of the temperature T is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally. This can be done either by explicitly giving a temperature or by selecting a temperature variable from another physics interface.

USING COMMON MODEL INPUT

When the option **Common model input** is selected in a physics node, you can see or modify the value actually used by clicking the **Go To Source** button ( ). Doing that will move you the node in the Model Builder that is defining the value of the property. That location can be:

- The **Default Model Inputs** node under **Global Definitions**
- A **Model Input** node under **Definitions** > **Shared Properties** in the current component

If you want to create a model input value which is local to your current selection, click the **Create Model Input** button . This will create a new **Model Input** node under **Definitions** > **Shared Properties** in the current component, having the same selection as in the current node.

See also Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

MASS DENSITY AND VOLUME REFERENCE TEMPERATURE

All structural mechanics interfaces are formulated on the material frame. This means that the equations of motion are written for a certain volume in its initial configuration.

The preservation of mass requires that the mass density is constant. In a structural mechanics problem this means that the mass density must not change. If you are using a material in which the density has a temperature dependence, you must specify a specific temperature at which the value is evaluated. This is the *volume reference temperature*. Conceptually, you can consider this as the temperature at which the domain has the size in which it is drawn. In practice, the choice of reference temperature is seldom an issue, unless your application requires extreme precision. The density of a solid material has a rather slow variation with temperature, so in most cases it is sufficient to use room temperature as reference.

If any material in the model has a temperature-dependent mass density, the **Volume reference temperature** list will appear in the **Model Input** section of the material settings. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

All effects of volume change with temperature are incorporated through the thermal expansion effects.

See also

- Using Common Model Input.
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*.

Thermal Expansion

As the temperature changes, most materials react by a change of volume. For a constrained structure, the stresses that evolve even with moderate temperature changes can be considerable. The volume change can be represented a *thermal strain* ε_{th}, which produces stress-free deformations. For a linear elastic material, the constitutive law is

$$\sigma = C : (\varepsilon - \varepsilon_{\text{th}})$$

In the computations, the thermal expansion appears as a load, even though it formally is a part of the constitutive relations.

You can include thermal expansion in a model either by adding a Thermal Expansion (for Materials) subnode to the chosen material, or by using the Thermal Expansion.

TEMPERATURE DEPENDENCE OF THE THERMAL EXPANSION COEFFICIENT

When performing an analysis over a larger range of temperatures, you often need to consider the temperature dependence in the values of the thermal expansion coefficient itself.

As long as you are using materials from the COMSOL Material Library, everything is handled internally. When you want to enter data from your own measurements or from the literature, you do, however, need to be aware of some details in the definitions used.

Tangent or Secant Data

Thermal expansion coefficients can appear in two forms: *tangent* and *secant*.

The tangent form states that the increment in length is

$$\frac{dL}{L} = \alpha_t(T)dT \tag{2-20}$$

where α_t is the tangential thermal expansion coefficient. This form, which is the thermodynamic definition, is conceptually simple, because α_t is uniquely defined at each temperature. It is, however, less convenient to use in practice because an integration is required for determining the actual change in length for a finite temperature difference.

The secant formulation, which is the default in COMSOL Multiphysics, is often used in engineering:

$$\frac{\Delta L}{L_0} = \alpha(T)\Delta T$$

In the secant formulation, the actual values of α will however depend on the choice of reference temperature, T_{ref}, at which the material has the reference length L_0:

$$\frac{\Delta L}{L_0} = \alpha(T,T_{\text{ref}})(T - T_{\text{ref}})$$

Converting from Tangent Form to Secant Form

Equation 2-20 can be integrated, giving

$$\ln\left(\frac{L}{L_0}\right) = \int_{T_{\text{ref}}}^{T} \alpha_t(\tau)d\tau \tag{2-21}$$

Define

$$I(T,T_{\text{ref}}) = \int_{T_{\text{ref}}}^{T} \alpha_t(\tau)d\tau$$

Thus,

$$\frac{\Delta L}{L_0} = \frac{L}{L_0} - 1 = e^{I(T,T_{\text{ref}})} - 1$$

giving the secant thermal expansion coefficient as

$$\alpha(T,T_{\text{ref}}) = \frac{e^{I(T,T_{\text{ref}})} - 1}{(T - T_{\text{ref}})}$$

For most materials and temperature ranges $I(T,T_{\text{ref}}) \ll 1$, which makes it possible to approximate with the simpler expression

$$\alpha(T,T_{\text{ref}}) = \frac{I(T,T_{\text{ref}})}{(T - T_{\text{ref}})} \tag{2-22}$$

If you have access to tangent data, you can choose between two different methods for using them in COMSOL Multiphysics:

- In most of the physics interfaces, you can enter tangent data directly by selecting **Tangent coefficient of thermal expansion** in the settings for **Thermal Expansion**. When using this option, a numerical integration of Equation 2-21 will be performed each

time the thermal strain is used. This will have a negative impact on the performance, when compared to using a secant coefficient of thermal expansion.

- Precompute the expression in Equation 2-20 externally for the intended range of temperatures. This can for example be done in a spreadsheet program. Enter the computed result as a function, which is then used as any other secant temperature dependent thermal expansion coefficient.

When using **Tangent coefficient of thermal expansion**, the integrate operator is used. It is called with the two integration limits being the reference temperature `<phys>.Tref` and the current temperature `<phys>.T`, where `<phys>` is the tag of the physics interface. If you define the expression for the coefficient of thermal expansion yourself, you must ensure that it depends on a 'free' variable, and not use the same temperature variable as you use to prescribe the current temperature `<phys>.T`.

Thermal Expansion Coefficient Dependence on Reference Temperature

Let $\alpha_m(T)$ be the temperature-dependent function that represents the measured values of the secant thermal expansion coefficient. The change in length of a sample at a given temperature T with respect to the sample's original length at a temperature T_m is called *dilation*.

Note that by definition, the dilation at $T = T_m$ is zero, so T_m denotes the strain-free state of the material as far as the measured values of $\alpha_m(T)$ is concerned. Denote the length of the sample at a temperature T as $L(T)$ and the strain-free length as $L_0 = L(T_m)$. The dilation can be then expressed as $L(T) - L(T_m)$. Using the definition of the secant coefficient of thermal expansion, $L(T)$ can be written as

$$L(T) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \tag{2-23}$$

When using the measured data, it is possible that the strain-free state occurs at a temperature T_{ref} which differs from T_m. The dilation at any temperature T would then be defined as $L(T) - L(T_{ref})$, where $L(T_{ref})$ can be written as

$$L(T_{ref}) = [1 + \alpha_m(T_{ref})(T_{ref} - T_m)]L(T_m) \tag{2-24}$$

As a result of this shift in the strain-free temperature, it is necessary to redefine the thermal expansion coefficient so that $L(T)$ and $L(T_{ref})$ can be related using Equation 2-23 but with T_m replaced by T_{ref}.

$$L(T) = [1 + \alpha_r(T)(T - T_{ref})]L(T_{ref}) \quad (2\text{-}25)$$

Here $\alpha_r(T)$ is the redefined thermal expansion coefficient, based on T_{ref}. It can be derived from the relations above. Using Equation 2-23 and Equation 2-25 there are two ways of writing the current length $L(T)$, so that

$$[1 + \alpha_r(T)(T - T_{ref})]L(T_{ref}) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \quad (2\text{-}26)$$

Equation 2-24 makes it is possible to eliminate $L(T_{ref})$ and $L(T_m)$ from Equation 2-26:

$$[1 + \alpha_r(T)(T - T_{ref})][1 + \alpha_m(T_{ref})(T_{ref} - T_m)] = 1 + \alpha_m(T)(T - T_m) \quad (2\text{-}27)$$

It is now possible find $\alpha_r(T)$, expressed in known quantities. After some algebra, the final expression is

$$\alpha_r(T) = \frac{\alpha_m(T) + (T_{ref} - T_m)\dfrac{\alpha_m(T) - \alpha_m(T_{ref})}{T - T_{ref}}}{1 + \alpha_m(T_{ref})(T_{ref} - T_m)} \quad (2\text{-}28)$$

In order to arrive at this form of $\alpha_r(T)$, the numerator has been rewritten, using

$$\begin{aligned}&\alpha_m(T)(T - T_m) - \alpha_m(T_{ref})(T_{ref} - T_m) = \\ &\alpha_m(T)(T - T_{ref}) + \alpha_m(T)(T_{ref} - T_m) - \alpha_m(T_{ref})(T_{ref} - T_m) = \\ &\alpha_m(T)(T - T_{ref}) + (T_{ref} - T_m)(\alpha_m(T) - \alpha_m(T_{ref}))\end{aligned} \quad (2\text{-}29)$$

Representation in COMSOL Multiphysics

Most materials listed in the material libraries and databases available with COMSOL Multiphysics and its add-on products contain a function for the measured temperature-dependent thermal expansion coefficient curve. You can find this from the Materials branch, as shown in Figure 2-29. The Piecewise function named `alpha_solid_1` is the measured thermal expansion coefficient $\alpha_m(T)$.

Using Functions in Materials in the *COMSOL Multiphysics Reference Manual*

The Material Contents section in Figure 2-29 shows the material property `alpha`, which is the redefined thermal expansion coefficient $\alpha_r(T)$. The complete expression for `alpha` is as follows:

```
(alpha_solid_1(T[1/K])[1/K]+(Tempref-293[K])*
if(abs(T-Tempref)>1e-3,(alpha_solid_1(T[1/K])[1/K]
-alpha_solid_1(Tempref[1/K])[1/K])/(T-Tempref),
d(alpha_solid_1(T[1/K]),T)[1/K]))/
(1+alpha_solid_1(Tempref[1/K])[1/K]*(Tempref-293[K]))
```

This is essentially Equation 2-28, but with a small modification to avoid problems if T=T_{ref}.

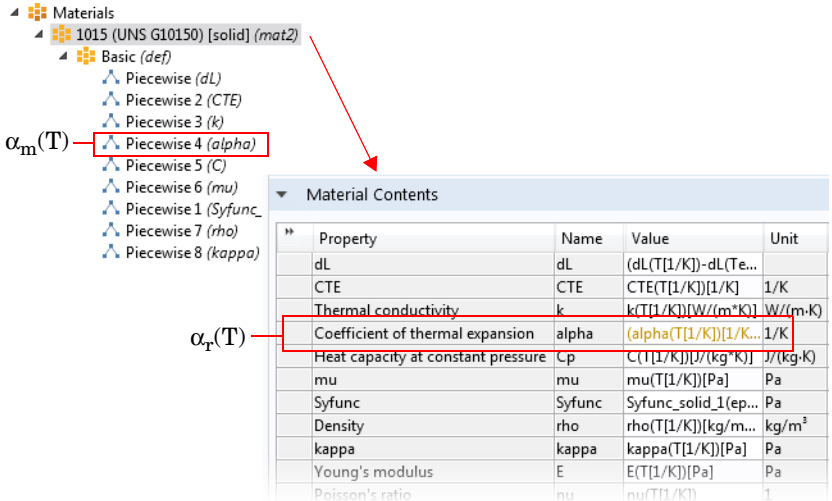


Figure 2-29: An example in COMSOL Multiphysics showing the Materials branch and where to find the temperature-dependent thermal expansion coefficient.

In the definition of `alpha` (to be more specific: `<material_tag>.def.alpha`) in COMSOL Multiphysics, T_{m} is set as 293 K and T_{ref} is obtained from the variable `Tempref`, which typically fetches its value from the physics interface.

Using Your Own Material Data

If you use our own material data in COMSOL Multiphysics (via an interpolation or any other function), you can still copy-paste the built-in expression for alpha into your New Material. You just need to:

- Replace the function name `alpha_solid_1(T[1/K])[1/K]`, with the function name that you have assigned to the temperature-dependent measured thermal expansion coefficient and use the correct temperature units.

Take care when describing the units. Temperature unit conversions can be the cause of subtle errors because of the shift in zero-point value. Use kelvin (K) as the temperature unit to the largest possible extent. As an alternative, you can use the other absolute temperature scale, Rankine (R). Avoid using Fahrenheit and Celsius unless you are completely familiar with how the temperature unit conversion works.

- Ensure that the value of T_{m} is changed from 293 K to the actual value of temperature that was used as the strain-free temperature to compute $\alpha_{\mathrm{m}}(T)$ in the function that you created.

User-Defined Materials and Libraries in the *COMSOL Multiphysics Reference Manual*

Constraints and Thermal Expansion

When a structure is undergoes thermal expansion, the fact that the motion is restricted by constraints will in general cause stresses. There are three types of such effects:

- The global expansion is restricted by constraints at different locations, so that internal forces are introduced throughout the structure.
- At a boundary that is constrained, local stresses can appear if the boundary is not free to expand in the tangential direction.
- Internally, the same type of local constraint effects will be caused by rigid objects, such as **Rigid Material**.

In many cases, not only the structure which actually is modeled deforms due to the changes in temperature, but also the surroundings (which are approximated by constraints) will deform. You can take this effect into account by adding a **Thermal**

Expansion subnode to the constraints. The constraints will then provide an extra displacement based on a given temperature field. For thermal strains, which have a simple variation in space (for example, linear temperature variations), it is possible to completely offset the constraint stresses using this method. For more general cases, the stresses caused by the constraint can be significantly reduced.

The thermal expansions of the constraints are independent of that of the material in the adjacent domain, so that the surrounding structure can be made from another material, or have a different temperature distribution.

You can, however, also inherit temperature and thermal expansion coefficient from the domain being constrained. This is useful for the common case that the temperature and materials are the same over the modeled structure body and its surroundings.

The spatial variation of the temperature and coefficient of thermal expansion must be explicit functions of the material frame coordinates. It is not possible to use a computed temperature distribution for the thermal expansion of the constraints.

- Thermal Expansion (for Constraints)
- Thermal Expansion of Constraints in the theory section.

For an example showing how to relive the stress at constraints in a heated structure, see *Thermal Expansion in a MEMS Device*: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/thermal_expansion**.

Thermoelastic Damping

In most engineering problems, the coupling between temperatures and structural problems can be considered as unidirectional. Only the thermal expansion is considered.

The opposite effect, where changes in stress cause heat generation, may be important in small structures vibrating at high frequencies. The **Thermoelasticity** interface, available with the MEMS Module, is designed for analyzing such problems.

It is also possible to take this effect into account by adding the **Thermoelastic Damping** node to the Heat Transfer in Solids interface. When you add a **Thermal Expansion** node to a material in the Solid Mechanics interface, the heat source term is computed and made available to the Heat Transfer in Solids interface.

When you add a **Thermal Expansion** node under the **Multiphysics Couplings** branch, it is possible to select whether the thermoelastic damping effect should be considered or not. The heat source contribution is then included automatically without adding any data in the heat transfer interface.

In either case, the heat source term is only present when Structural Transient Behavior is set to **Include inertial terms**.

See also

- Entropy and Thermoelasticity
- Thermal Expansion (for Materials)
- The Fluid–Solid Interaction Interface

Fluid–Structure Interaction

Fluid–structure interaction (FSI) involves several phenomena where a solid structure and a fluid interact with each other. The interaction has three possible components, which can be more or less important:

- The pressure and viscous forces in the fluid provides a load on the boundary of the solid. Usually, the pressure is dominant.
- The deformation of the structure changes the geometry of the fluid domain.
- The fluid sees the structure as a moving wall, which imposes a velocity at the interface.

You can model FSI with five different structural mechanics interfaces: Solid Mechanics, Multibody Dynamics, Shell, Layered Shell, and Membrane. The fluid flow can be modeled with any domain-level physics interface from the **Single-Phase Flow** group and the **Two-Phase Flow** groups under **Multiphase Flow**.

Deforming Fluid Domains

When a Fluid–Structure interaction multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** node can be added automatically in the **Model Builder** under the **Definitions** node. This is the case for the following interfaces:

- Fluid–Solid Interaction
- Fluid–Shell Interaction
- Fluid–Membrane Interaction
- Fluid–Multibody Interaction
- Fluid–Multibody Interaction, Assembly

The **Deforming Domain** node is, however, not added for multiphysics interfaces denoted 'Fixed Geometry', which are intended for cases where the deformation of the fluid domains is small everywhere.

A deforming domain represents domains and boundaries where the mesh can deform. By default, the **Deforming Domain** node has an empty selection. You can then select any fluid domain. However, this is only needed if the geometry of such a domain experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could introduce unnecessary

overhead and nonlinearity in the model (see Deformed Mesh Fundamentals). Domains selected in the Solid Mechanics interface are not applicable. The deformations of the solid parts are handled in the formulation of the Solid Mechanics interface.

If no domains are selected in the **Deforming Domain** node, a fixed geometry case is solved.

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*

By default, the mesh is free at all exterior boundaries of the geometry, and it follows the solid boundaries at the Solid–fluid interfaces. You can also add other types of boundary conditions for the mesh motion; for details, see Deformed Geometry and Moving Mesh in the *COMSOL Multiphysics Reference Manual*.

Union or Assembly

In most cases you model FSI problems so that the geometry sequence is set up to form a union, and the same multiphysics coupling, Fluid–Structure Interaction, is used irrespective of the type of structural mechanics interface. This coupling will automatically find all boundaries that are shared between the structure and the fluid.

There are, however, cases where the assembly mode must be chosen, particularly when having mechanisms, as is common in the Multibody Dynamics interface. In that case, the interface between the solid and the fluid is no longer formed by a common boundary. Rather, it consists of two boundaries, located at the same place in space. These boundaries will in general slide with respect to each other. To model this, you use the Fluid–Structure Interaction, Pair multiphysics coupling. You must create appropriate pairs containing the boundaries from both types of physics under **Definitions**, and manually select them in the **Fluid–Structure Interaction, Pair** node.

UNION — FLUID–STRUCTURE INTERACTION

When using **Fluid–Structure Interaction**, the spatial frame also deforms with a mesh deformation that is equal to the displacements `u_solid` of the solid within the solid domains. The mesh is free to move inside the fluid domains, and it adjusts to the motion of the solid walls. This geometric change of the fluid domain is automatically accounted for in COMSOL Multiphysics by the ALE method.

ASSEMBLY — FLUID–STRUCTURE INTERACTION, PAIR

The **Fluid–Structure Interaction, Pair** coupling does not automatically transfer the mesh deformation at the interface from structural displacements.

For the moving mesh, you must specify the deformation of the mesh manually. Add a **Prescribed Mesh Displacement** node under the **Moving Mesh** node, in which you give the structural displacement as the mesh displacement expression. The variables for the displacement in the structure is provided by the multiphysics coupling.

The variable names to use have the form *<tag>*`.u_solid`, *<tag>*`.v_solid`, and *<tag>*`.w_solid`, where *<tag>* is the tag of the multiphysics coupling, for example `fsip1`.

Contact and FSI

Sometimes the structural deformations are so large that objects may come into contact with each other with the fluid being squeezed in between. Modeling contact together with FSI requires some special considerations. The mesh in the fluid domain may deform, but the topology remains the same — a fluid domain cannot be split into two. If you are to model a valve or a similar structure, then the two solid parts cannot come exactly into contact.

By adding an offset in the settings for the **Contact** node, you can force the two sides of the solid to experience contact at some distance before they meet in the geometrical sense. This approach only will, however, leave a thin channel through which the fluid can pass. The reduction in flow may be sufficient, but you can block it even further by increasing the viscosity in the channel when the gap is closed. To do that, you can, for example, compute the minimum gap anywhere in the contact pair, and then make the viscosity a function of it. Another option is to compute the wall distance in the fluid from both sides of the contact pair and use that information to modify the viscosity. Do not increase the viscosity more than a couple of orders of magnitude, to avoid numerical problems.

In configurations where you more or less completely cut off the whole flow, you must pay particular attention to your boundary conditions. A prescribed flux will cause an extreme pressure buildup upstream of the valve and thus unrealistically large forces on the structure.

One-Way FSI

For some fixed geometry FSI problems, you can consider the coupling as being unidirectional. One physics interface affects the other, but it is not reciprocal. Typical examples are when the fluid slightly deforms the structure, or when small structural vibrations modify the fluid flow. In this case, it is not necessary to compute the solution with both physics interface solved together. It is more efficient to first solve for the governing physics interface only, and then the other one with results from the first as input. To solve such a problem sequentially, you need to create a study configuration manually, with one study step for each physics interface. The governing physics depends on the coupling type. For the case of fluid loading on a structure, the governing physics is the fluid one. For the case of velocity transmission to a fluid, the governing physics is the solid one.

Deformed geometry FSI, or fixed geometry fully coupled FSI, should not be solved using a one-way approach.

Below are the steps to follow to compute a one-way FSI problem sequentially:

1. In the study step settings windows, under the **Physics and Variables Selection** section, clear the physics that is solved in the second step, so that only the governing physics is selected.
2. Add a second study step to the study and in the settings windows make sure you have the governing physics cleared.

   If the selected study steps are of stationary type, you can generate the default solver configuration, edit it if necessary, and compute the solution. The mapping of the solution from the first to the second study step is done automatically.

   In case of a transient problem, continue with the steps below:
3. In the second study step settings window, expand the section **Physics and Variables Selection**. Under **Initial values of variables solved for** make sure the settings are defined as in the table below:

Settings	User controlled
Method	Solution
Study	<previous study step>
Selection	First

4 Under **Values of variables not solved for** define the settings as in the table below:

Settings	User controlled
Method	Solution
Study	<previous study step>
Selection	All

In the case of a fluid loading to structure coupling type, the structural mechanics problem can be treated as quasistatic. This can be handled by running the structural analysis as a parametric sweep over a number of static load cases, where time is used as the parameter.

For an example of one-way FSI using a quasistatic structural analysis, see *Fluid–Structure Interaction in a Network of Blood Vessels*: Application Library path **Structural_Mechanics_Module/Fluid-Structure_Interaction/blood_vessel**.

Component Mode Synthesis

Introduction

Component mode synthesis (CMS) is a family of procedures in which one or more parts of a finite element model are represented by a small reduced-order model (ROM). There are several possible algorithms through which such a reduction can be made. In COMSOL Multiphysics, the popular *Craig–Bampton method* is used for model reduction with a dual assembly technique to ensure compatibility between components.

The reduced component is sometimes called a *dynamic substructure*. Essentially, it contains small mass, damping, and stiffness matrices, tuned so that it can represent the static stiffness, as well as a set of important eigenmodes and the inertia of the original component. An important assumption is that the reduced structure is linear.

The reduced component accounts for the constraints applied on the component. It can also contain, for example, information about loads applied to the component prior to model reduction.

Since the reduced components only have a small number of degrees of freedom (often of the order of 10–100), they are computationally more efficient than the original full FEM components. To create a reduced component, it is necessary to both perform an eigenfrequency analysis and to solve for a number of static load cases. These studies are, however, computed at the component level, and are thus usually computationally much cheaper than analyzing the full model.

For general information on reduced-order modeling and model reduction, see also

- Reduced-Order Modeling
- Model Reduction

in the *COMSOL Multiphysics Reference Manual*.

COMSOL Implementation

The Component Mode Synthesis (CMS) technique is currently implemented in the Solid Mechanics, Shell, and Multibody Dynamics interfaces. You use the **Reduced**

Flexible Components node in either of these interfaces to control the generation and use of such reduced components.

Reduced components are connected to each other and to nonreduced parts of the model through Attachment features. It means that all physics features that can access an attachment can be used to connect the reduced components. The most common option is to use joints, but there are other alternatives, such as **Spring–Damper**, or bearing foundations in the rotordynamics interfaces.

The global model can consist of either a set of reduced components, or a mix of ordinary FE discretized domains and reduced components. The reduced components can be used as linear parts in an otherwise nonlinear model.

A requirement to be able to create a valid reduced component is that its underlying physics is linear with regards to material models, the strain formulation, and other possible contributions such as springs and loads

	COMSOL Multiphysics always forces a geometrically linear formulation on selections that intersect that of a **Reduced Flexible Components** node.

	When using a reduced component in the global model, the connection to nonreduced parts must be such that the deformation of the reduced component is limited to small rotations and small strains.

You can compute all type of results and visualize reduced components just as if they are ordinary domains in your model. One important property of the reduced component is that only the results for the reduced set of degrees of freedom are stored. It means that for time-dependent studies with many time steps, the file sizes can be reduced by orders of magnitude.

Applications

Some examples of applications of the CMS reduction technique include:

- In multibody dynamics models, you usually have a number of physical components, coupled by joints. The ultimate reduction is then to use rigid domains. When it is necessary to take the flexibility of a component into account, using a reduced flexible component is attractive.

- In rotordynamics, it is common that the foundation supporting the rotor must be part of the analysis. Such a foundation will almost invariably have linear properties. It is then well suited to be considered as a reduced flexible component. This can greatly reduce the number of the degrees of freedom in the rotordynamic analysis, which is often computationally heavy.
- In many situations, only a smaller part of a model is nonlinear. It is then possible to reduce the linear parts once and for all, and then iterate on a much smaller nonlinear system of equations. For this to be feasible, it must however be possible to dissect the structure in a way that can be represented by attachments.

Working with CMS Models

CREATING REDUCED COMPONENTS

A reduced component can be created in an applicable physics interface by adding a **Reduced Flexible Components** node, in which you select all domains that are to be reduced, as shown in Figure 2-30. Subsequently, one **Component Definition** subnode should be added for each set of domains for which a reduced-order model (ROM) is to be generated. By default, this step is automated and the selected domains are grouped into disconnected components as detected from the geometry. With this setting, a number of **Component Definition** subnodes are created and their selections are set automatically. However, by setting **Component definition** to **User defined** in the **Reduced Flexible Components** node, you can take manual control of the geometric definition of the components by adding, removing, or modifying **Component Definition** nodes. In each such node, select a number of domains that defines a component. This may be necessary if domains are not physically adjacent, but connected by other means, for example, by springs.

By default, disconnected geometries connect by the **Continuity** and **Thin Elastic Layer** pair features are merged when automatically generating **Component Definition** subnodes. This also applies to **Boundary to Boundary**, **Edge to Boundary**, and **Edge to Edge** connection features in the Shell interface. The automatic handling of such connections can be disabled by clearing **Include connections and pairs in component definition** in the **Reduced Flexible Components** node.

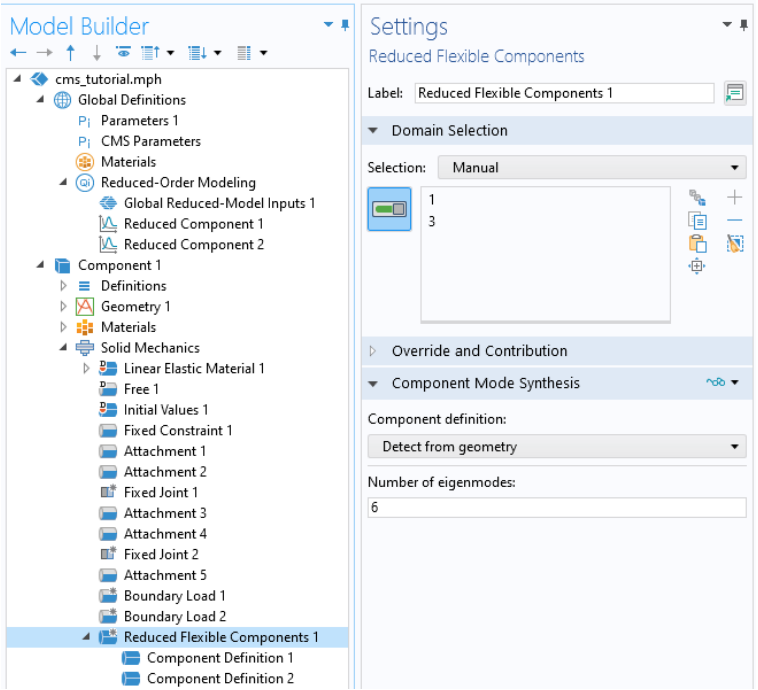


Figure 2-30: A Reduced Flexible Components node with two automatically generated Component Definition subnodes.

- If the components are geometrically disjoint, then **Form Union** should be used in the geometry sequence. If not, **Form Assembly** is probably a better choice, but then you may have to either create unions between domains that are part of the same components, or edit the settings in the **Component Definition** nodes manually.
- In the majority of cases, it is sufficient with one **Reduced Flexible Components** node per physics interface. If you, however, do not want to train all ROMs in one sweep, having a single **Reduced Flexible Components** node per actual component can be useful. Another case is if you decide to reduce one additional component after already having reduced one or several other components. Then, adding an additional **Reduced Flexible Components** node can be an alternative.

	The **Reduced Flexible Components** node is only applicable on domains where the material behavior is determined by a **Linear Elastic Material** node or a **Section Stiffness** node in the Shell interface. You can, however, have several such nodes with different properties and settings within one reduced component.

A disconnected reduced component, as defined by a **Component Definition** subnode, can only be connected to other parts of the geometry by attachment features. In fact, to be able to create the ROM, each component must be connected to at least one **Attachment** node. By using attachments, a reduced component can be connected to any number of other parts, reduced or full, of the model assembly.

	See also Attachments.

When working with CMS, the full static and dynamic behavior of a component is represented by a number constraint modes and constrained eigenmodes that are computed in the training phase. These are independent of each other and are used to construct one ROM for each component. Each **Attachment** node connected to a component will add a number of static load cases that describe the constraint modes: six in 3D, and three in 2D. The number of eigenmodes to be used is controlled manually. You can either define it for all components in the **Reduced Flexible Components** node, or individually in each **Component Definition** subnode. To get an accurate representation of the reduced component, always make sure to use a sufficient number of eigenmodes to describe the dynamics of each component. During the eigenfrequency training step, all attachments are treated as fixed constraints, hence, the eigenmodes are always constrained.

Note that, from a computational point of view, each eigenmode is represented as a degree of freedom when using the reduced component in a global analysis. For computational efficiency, you should avoid using unnecessarily many eigenmodes.

To create the ROMs, a special study sequence needs to be set up for each **Reduced Flexible Components** node. It should sweep over all **Component Definition** subnodes, and for each component, compute the static load cases and the requested eigenmodes in training study steps as outlined above. The results from these study steps are then used in a **Model reduction** step to generate a ROM. By using the **Configure CMS Study** ( ) button in the **Reduced Flexible Components** node, the setup of this special study

sequence is automated. In the model tree, a generated ROM is represented as a node with the label **Reduced Component** under **Global Definitions** as seen in Figure 2-31.

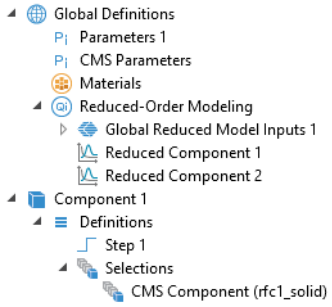


Figure 2-31: Nodes in the model tree that are added automatically when working with CMS.

- When a **Reduced Flexible Components** node is added, a set of parameters are also automatically added under the **Global Definitions** branch. These are placed in the **CMS Parameters** node seen in Figure 2-31, which is created by the first **Reduced Flexible Components** node in the model. Additionally, each **Reduced Flexible Components** node creates an explicit selection node with the label **CMS Component (<tag>)** in its model component. The parameters and selection are used in the corresponding CMS study and should therefore not be deleted or modified. As a safeguard, they are regenerated if missing when the **Configure CMS Study** ( ) action is executed.
- If you change the number of **Component Definition** nodes or add new features to the physics interface after setting up the CMS study and want to run it again, the safest option is to reconfigure the study by using the **Configure CMS Study** ( ) action. Note that this will delete any ROMs that have previously been generated. To save these in the model, you can temporarily change the name of the existing **Reduced Component** nodes under **Global Definitions** before reconfiguring the CMS study.

APPLYING LOADS TO REDUCED COMPONENTS

Loads can be applied to a reduced component in two fundamentally different ways:

1. External to the ROM: In this case, the weak contribution of the load on the reduced component is part of the assembly procedure for the global FE model. This is the default choice and can handle any type of loads. It is, however, the computationally more expensive of the two methods.
2. Internal to the ROM: The load can either be constant or a function of a global reduced model input. In this case, the load is stored as part of the ROM.

Be careful not to mix up the two alternatives; doing so can lead to double load contributions. For this reason, option 1 is usually recommended. When the **Configure CMS study** ( ) button is used, all load type features are disabled in the study steps of the CMS study.

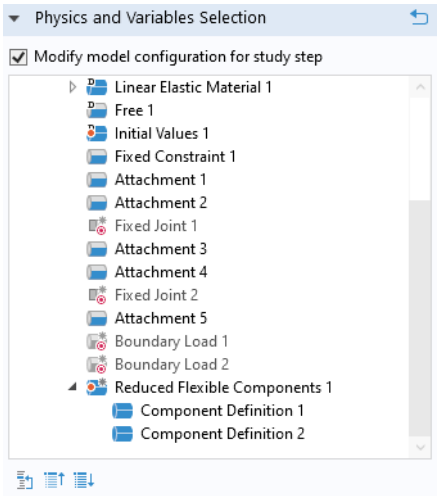


Figure 2-32: Loads are, by default, disabled in the model reduction study.

The advantage of option 2 is that it can be computationally cheaper in the global analysis, since any evaluation of weak equations in domains of the reduced component is avoided. To use this approach, enable the relevant load features in the reference study step to the **Model Reduction** study step in the CMS study. The same load features should then be disabled in all other studies. By selecting **Include load contributions in Reduced Components** in the **Reduced Flexible Components** node, you can change the

default behavior of loads when the **Configure CMS study** ( ) button is used. When selected, all loads are enabled in the study steps of the generated CMS study.

You can find an example of both approaches for load application in this tutorial model:

- *Component Mode Synthesis Tutorial*: Application Library path **Structural_Mechanics_Module/Tutorials/cms_tutorial**
- *Component Mode Synthesis Tutorial*: Application Library path **Multibody_Dynamics_Module/Tutorials/cms_tutorial**
- *Component Mode Synthesis Tutorial*: Application Library path **MEMS_Module/Dynamics_and_Vibration/cms_tutorial**

While conceptually similar to other load features, **Linearly Accelerated Frame**, **Rotating Frame, Gravity**, and **Base Excitation** are always considered as internal to the ROM and part of the reduced component. They are, by default, enabled in the **CMS Study** when using the **Configure CMS study** ( ) button. The reason is that they are considered to contribute to the acceleration of the frame. If necessary, use control inputs to parameterize, for example, the gravity vector or rotation speed.

THE CMS STUDY

The CMS study is a parametric sweep over a set of components as defined by the **Component Definition** subnodes of a **Reduced Flexible Components** node with the aim to create one ROM for each component. The study is automatically generated by using the **Configure CMS Study** ( ) button in a **Reduced Flexible Components** node. The auto-generated study and solver sequences have a number of built-in features and settings that greatly simplifies the process of creating a reduced component; especially when there are multiple components to be reduced. The study sequence is shown in Figure 2-33 and consists of:

- A **Parametric sweep** over the components to be reduced
- A **Stationary** training study step to compute the constraint modes from the static load cases
- An **Eigenfrequency** training study step to compute the constrained eigenmodes
- A **Model Reduction** study step to generate the ROM. This step can either use a **Time Dependent** or **Frequency Domain** study step as a reference.

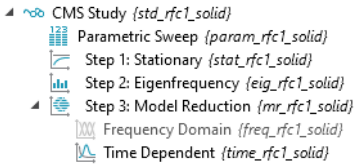


Figure 2-33: The automatically generated CMS study.

If the global model is to be used in a frequency domain analysis, use a **Frequency Domain** study step as a reference during model reduction. If not, use a **Time Dependent** study step. Both steps are added by the **Configure CMS study** ( ) action, and can be enabled or disabled depending on the use case. By default, the **Time Dependent** study step is enabled.

Key to the CMS study is that all study steps are solved on a subset of the total model; the selection of the component. This is controlled in the **Compile Equations** nodes of the solver sequence. When using an auto-generated CMS study, the selections used are automatically updated during the sweep over components.

Running a CMS study creates a number of datasets in the **Results** branch as well as subnodes to the **Reduced-Order Modeling** node under **Global Definitions**.

- The **Reduced Model Data** datasets contains the solution that is used to create a ROM, including the constraint modes and eigenmodes. It also contains the matrices of the ROM, which can be inspected by using the **System Matrix** derived values node.
- Depending on the chosen reference study step during model reduction, **Frequency Domain, Modal Reduced-Order Model** or **Time Dependent, Modal Reduced-Order Model** nodes are created under **Global Definitions**. When created from a CMS Study, these are always created with a default label **Reduced Component** and names that are related to the generating study and physics. Moreover, the CMS study generates ROMs with a stateful interface, which is a requirement for it to connect to other parts of the model. You can use **Model Control Inputs** and other settings in the generated ROMs to modify their behavior when used in a global analysis.

Once the ROMs have been created, you can in principle delete the CMS study to clean up the model tree. In fact, you only need to keep the **Reduced Flexible Components** node and the generated ROMs under **Global Definitions**.

Neither the names, nor the order of the list of ROMs in the **Reduced-Order Modeling** node under **Global Definitions** should be changed, as this will break the connection between the ROMs, the physics, and the reduced component. The automatic naming convention has the structure `<rom>_n_<feat>_<phys>_<i>`, where `<rom>` is a generic tag for the generating ROM, `<feat>` is the tag of the **Reduced Flexible Components** node that generated the CMS study, `<phys>` is the physics tag, and `<i>` is a number. For example, the ROM of the first component in a Solid Mechanics interface is typically named `rom1_n_rfc1_solid_1`.

SETTING UP THE GLOBAL MODEL

Using ROMs created from a **Reduced Flexible Components** node and a CMS Study in a global model, in general, requires no further steps to be taken. The only requirement is that the states of the ROMs are solved together with the other dependent variables of the model. By default, the **Solve for** status of these ROMs are synchronized with the **Solve for** status of the generating physics in non-CMS studies, which takes care of this requirement.

If all the domains of the physics interface are reduced, it is possible to turn off the synchronization of the **Solve for** status, since in such cases it is only necessary to solve for the states of the ROMs. By not solving for the dependent variables of the physics, it is made sure that no double contributions are added by, for example, load features. Clear the **Synchronize 'Solve for' study setting for Reduced Components** checkbox in the **Reduced Flexible Components** node to make this possible.

One way to add more control over the behavior of a ROM is to add control inputs in **Global Reduced Model Inputs** under **Global Definitions**. These should be added to the model before solving the CMS study, and can be used in expressions in relevant physics nodes. After running the CMS study, you can modify the input expression for each

added control input, either globally or individually in each of the generated **Reduced Component** nodes.

If you include a load in a reduced component, and enter a value or a parameter for its amplitude, then that load will have a constant value when the component is reused. If you want to be able to, for example, make that load time dependent, or to switch it off, then you define it using a control input.

For a more detailed discussion on how to work with ROM control inputs and their limitations, see Reduced-Order Model Inputs in the *COMSOL Multiphysics Reference Manual*.

IMPORTANT CONSIDERATIONS

A number of important considerations to be aware of when working with CMS and reduced components are listed below:

- Do not modify the content of automatically generated nodes in the model tree, such as the **CMS Parameters** node, and the **CMS Component** selection.
- Only zero-valued constraints can be applied to a reduced component. While it is fundamentally possible to apply constraints on a reduced component in a global analysis, the correct way is to apply them when generating the ROMs in a CMS study so that such constraints can be accounted for while computing constraint modes and eigenmodes. Later, constraints can be removed in all other studies. This is handled automatically for most such features including for example:
  - **Fixed Constraint**
  - **Prescribed Displacement**
  - **Roller**
  - **Symmetry**

  No constraint equations are added for these features on selections that intersect that of a **Reduced Flexible Components** node. For the **Prescribed Displacement** node and similar features, any nonzero values set for the constraint are, moreover, ignored in the CMS study.
- **Attachment** nodes can be used without a connection to other parts of the model to define additional static modes of the reduced component. Note that attachments by default induce a rigid boundary on its selection.

- **Added Mass**, **Spring Foundation**, and **Thin Elastic Layer** are special features that add either mass or stiffness to the reduced component, and can therefore be considered as part of its basic properties. By default, these are enabled in the **CMS Study** when using the **Configure CMS study** ( ) button. No mass nor stiffness is added in a global study on selections intersecting with that of a **Reduced Flexible Components** node.
- When working with CMS, the following subnodes to **Linear Elastic Material** and **Section Stiffness** are supported when creating reduced components:
  - **Thermal Expansion**
  - **Hygroscopic Swelling**
  - **Initial Stress and Strain**
  - **External Stress**
  - **External Strain**
  - **Damping**
  - **Fiber**

  All other subnodes are, by default, disabled in the **CMS Study** when using the **Configure CMS study** ( ) button. Some options in the supported features may, however, not be supported. For example, the **External material** option in **External Strain** is not supported with CMS.
- In the **Damping** subnode to **Linear Elastic Material** or **Section Stiffness**, the following **Damping types** are supported:
  - **Isotropic loss factor**
  - **Anisotropic loss factor**
  - **Loss factor**
  - **Rayleigh damping**
  - **Viscous damping**
  - **Wave attenuation**

  When using the **Configure CMS study** ( ) button, the **Damping** subnode is disabled in the training steps to avoid complex eigenpairs, but active in the reference step to the model reduction. Any contributions are removed on selections that intersects with that of a **Reduced Flexible Components** node in a global study.
- Damping can also be added to reduced components by **Spring Foundation**, **Thin Elastic Layer**, and **Low-Reflecting Boundary** nodes. For these features, damping contributions are only added on selections intersecting with that of a **Reduced**

Flexible Components node in the reference step to the model reduction. No contributions are added in the training steps for such selections to avoid complex eigenpairs.

- Reduced components are by definition linear. Do not use any features that are nonlinear such as **Creep**, **Damage**, or **Plasticity** on the same selection as a **Reduced Flexible Components** node. Also, make sure not to induce nonlinearity by user defined expressions in features that are to be reduced. This also applies to boundary, edge, and point features adjacent to the domains selected in a **Reduced Flexible Components** node.
- It is not possible to compute dissipated energy due to for example damping on selections that intersect that of a **Reduced Flexible Components** node. The **Calculate dissipated energy** setting is ignored for such selections.
- Certain features that add weak contributions on domain, boundary, edge, or point level should be used with care if their selections intersect that of a **Reduced Flexible Components** node. To get consistent results and avoid double contributions, it may be necessary to manually disable some features in the model tree of the training study step of a generated CMS study, or in the global study that uses a ROM. Examples of such features, other than loads, include **Weak Contribution**. By default, these are disabled in the **CMS Study** when using the **Configure CMS study** ( ) button.
- Most features that add global dependent variables are not applicable together with reduced components. The reason is that it is difficult to automatically determine to which part of the model such variables belong. Hence, do not use features such as **Average Rotation**, **Rigid Body Contact**, **Rigid Connector**, **Prescribed Velocity**, or **Point Load, Free** when setting up reduced components. By default, all such features are disabled in the **CMS Study** when using the **Configure CMS study** ( ) button. The only exception is the **Attachment** feature which is specially designed to work with reduced components.
- Features and functionality that add dependent variables which are expected to have a significantly different order of magnitude compared to the displacements should be used with caution when creating reduced components. The reason is that such DOFs can corrupt the scaling of the eigenvectors used to train the ROM during

model reduction, which in the end will result in an inaccurate ROM. Examples of functionality that may cause this are:

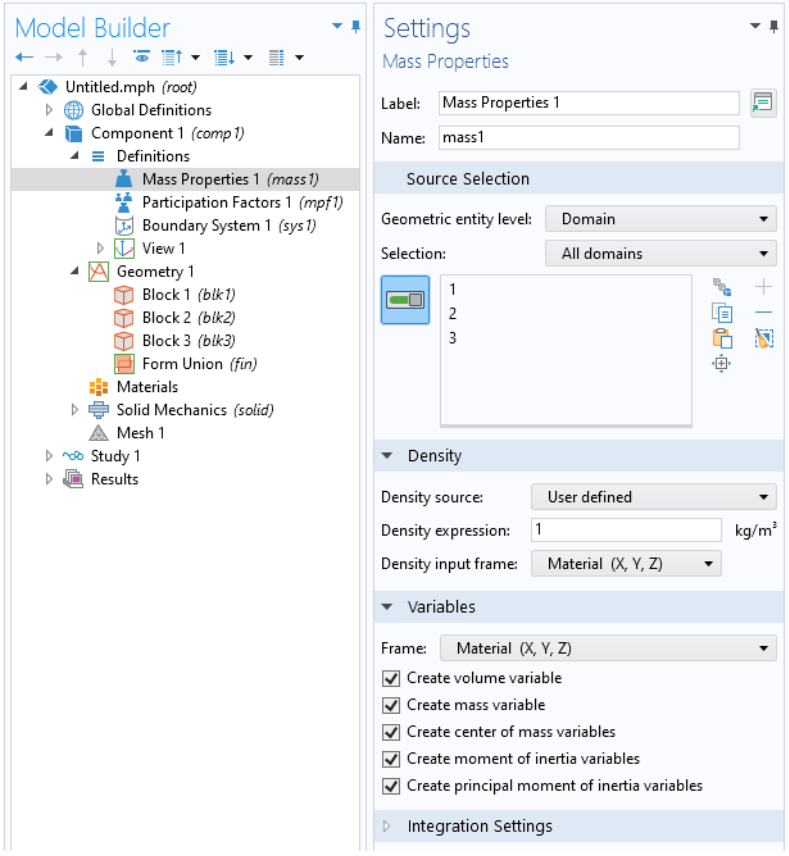
- The **Pressure** mixed formulation in **Linear Elastic Material**
- The **Flexible** connection type in **Attachment** nodes
- The Lagrange multipliers added by weak constraints

If such functionality must be used, you can try to manually adjust the scaling of the eigenvectors by changing the **Maximum absolute value** in the **Eigenvalue Solver** of the **CMS Study**. Another alternative is the try setting **Scaling of eigenvectors** to **Mass Matrix**.

- Translating or rotating a ROM is not supported. Hence you cannot use a single ROM to represent multiple reduced components.
- Import of ROMs created in another COMSOL Multiphysics model or any external software is not supported for reduced components.

Computing Mass Properties

In structural mechanics analysis, especially when modeling dynamic problems, the mass properties of a structure or its part can be an important aspect of the design. To compute such mass properties, you can use **Mass Properties** node, which can be added under **Component** > **Definitions** > **Physics Utilities**. There, you can select the geometry domains to be included into the computations and select physics interfaces that will define the mass properties. You can add and configure several mass property contributions if needed.



Volume, mass, center of mass, and moments of inertia will be computed. They will become available as predefined variables, which you find in the Equation View node's settings window under the corresponding **Mass Properties** node.

Variables

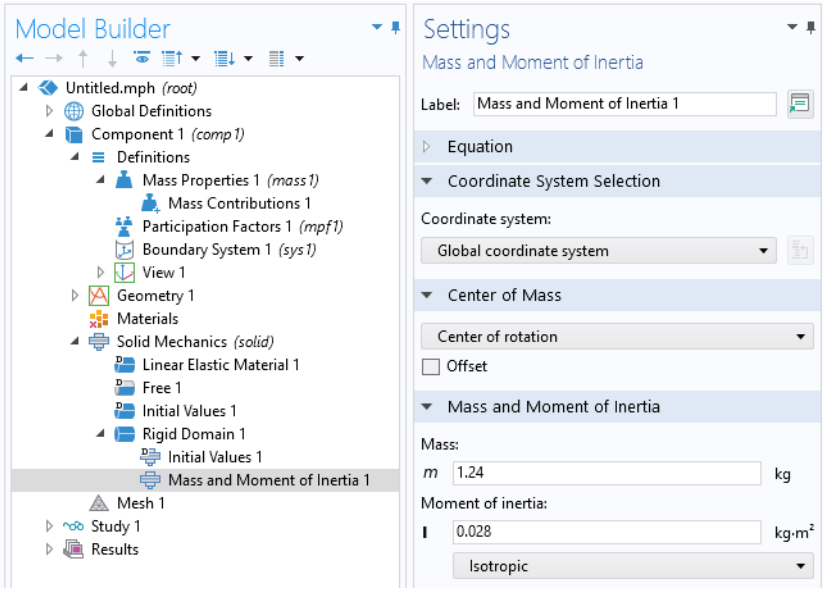
Name	Expression	Unit	Description
mass1.volume	mass1.intDim3(root....	m^3	Volume
mass1.mass	mass1.intDim0(mas...	kg	Mass
mass1.CMX	if(mass1.mass>0,(m...	m	Center of mass, X component
mass1.CMY	if(mass1.mass>0,(m...	m	Center of mass, Y component
mass1.CMZ	if(mass1.mass>0,(m...	m	Center of mass, Z component
mass1.IXX	mass1.intDim0(mas...	$kg·m^2$	Moment of inertia XX component
mass1.IXY	mass1.intDim0(-ma...	$kg·m^2$	Moment of inertia XY component
mass1.IYX	mass1.IXY	$kg·m^2$	Moment of inertia YX component
mass1.IXZ	mass1.intDim0(-ma...	$kg·m^2$	Moment of inertia XZ component
mass1.IZX	mass1.IXZ	$kg·m^2$	Moment of inertia ZX component
mass1.IYY	mass1.intDim0(mas...	$kg·m^2$	Moment of inertia YY component
mass1.IYZ	mass1.intDim0(-ma...	$kg·m^2$	Moment of inertia YZ component
mass1.IZY	mass1.IYZ	$kg·m^2$	Moment of inertia ZY component
mass1.IZZ	mass1.intDim0(mas...	$kg·m^2$	Moment of inertia ZZ component

Besides for postprocessing purposes, you can also use these variables in any user-defined expressions, user inputs, and in optimization criteria.

Structural mechanics interfaces contribute to the mass properties in several ways:

- All material models, including **Rigid Material**, define mass density contributions.
- **Added Mass** nodes for all geometric entity levels can also contribute with added mass density. It is possible to suppress the contribution from an **Added Mass** node by using the **Exclude contribution** checkbox in the **Frame Acceleration Forces** section.
- **Point Mass** nodes contribute both with mass and with the specified mass moment of inertia. It is possible to suppress the contribution from a **Point Mass** node by using the **Exclude contribution** checkbox in the **Frame Acceleration Forces** section.
- The **Hygroscopic Swelling** nodes, which can be added to most material models for all structural elements, can use the moisture content as an extra mass density contribution.

- **Rigid Material** and **Rigid Connector** can contribute directly to the total mass properties via their **Mass and Moment of Inertia** subnodes.



- Structural elements like beams and shells take their true geometrical dimensions into account when contributing to the mass properties. As an example, a beam contributes to the rotational inertia around its axis, even though the geometrical model is only an edge. The beam cross-section properties are used to compute the data.

The mass properties can be computed on both initial geometry (material frame) and deformed geometry (spatial frame). The results may differ considerably in case of large deformations. To compute the results in the undeformed geometry, you do not have to perform the whole analysis; it is sufficient to choose **Get Initial Values** under the **Study** node. To obtain the mass properties in the deformed configuration, you need to the full analysis, so that the displacement results are available.

In the *COMSOL Multiphysics Reference Manual*:

- Mass Properties
- Studies and Solvers
- Derived Values, Evaluation Groups, and Tables

Effective Properties of Periodic Structures

Sometimes you need to model on different scales. The heterogeneous properties of a material in the microscopic scale are often unfeasible to use directly on a macroscopic scale, in which a structure is typically analyzed. In such a case, you typically want to use a homogeneous material model with appropriately averaged properties.

The Cell Periodicity feature facilitates the evaluation of such average properties. It is based on the idea of a *representative volume element* (RVE). The RVE is a domain that is representative for the material on a microscopic scale.

If the material is truly periodic, an RVE is typically identified as the smallest possible unit cell. If the material has a random distribution of, for example, porosities, the RVE should be large enough to be representative for the average properties of the material on a macroscopic scale.

The only requirement on the shape of the RVE is that it should be possible to fill space with a repetitive pattern of RVEs. This means that there are a set of matching boundary pairs, each pair having the same geometry, but offset by a given distance.

Material Homogenization Using the Cell Periodicity Node

To model a periodic unit cell or a RVE (representative volume element), you add the domain feature **Cell Periodicity**, and select the domains representing the cell. For each pair of matching boundaries, add a **Boundary Pair** subnode, and select the corresponding boundaries, typically on the opposite sides of the cell.

In principle, there is no limitation on the physics features you can use for modeling the cell, as long as the basic assumptions about periodicity and homogeneity are valid. You should, however, not add any displacement constraints since the possible rigid-body motions are automatically constrained by the **Cell Periodicity** node.

MESHING

For periodic boundary conditions, the accuracy of the analysis is significantly better if each pair of matching boundaries has the same mesh. This can be accomplished for example using the Identical Mesh feature in the mesh sequence.

When using nonmatching boundary meshes, the stress disturbances can be reduced by using a weak constraint.

HOMOGENIZED MATERIAL PROPERTIES

Two different types of material properties can be automatically evaluated using the **Cell Periodicity** node. With the **Free Expansion** periodic condition, you can calculate the homogenized material properties related to the free expansion of the RVE when exposed to a variation in temperature or moisture concentration — that is, the averaged coefficient of thermal expansion and the averaged coefficient of hygroscopic swelling. To set up an analysis for the evaluation of thermal properties, add a **Thermal Expansion** subnode to each material model used in the RVE, and apply a one degree increment in temperature. Similarly, for hygroscopic properties, add **Hygroscopic Swelling** subnodes and apply a unit increment in concentration. Without adding these subnodes to the material models of the RVE, no expansion is applied to the unit cell, and consequently the results will be zero. The results will also be zero if no variation in temperature or concentration is applied. If requested, the computed material properties are by default presented in a separate evaluation group under **Results**.

With the **Average strain** or **Average stress** periodic conditions, you can calculate the homogenized elastic properties of the RVE. Use **Average strain** to compute the elasticity matrix, or **Average stress** to compute the compliance matrix. In order to compute these properties, the deformation of the RVE needs to be evaluated for a number of fundamental load cases. These correspond to the perturbations of the unit cell with each component of the average strain or stress tensor, while keeping the other components equal to zero. Setting up this analysis can be automated with the **Create Load Groups and Study** option under the **Study and Material Generation** button. It creates a load group for each tensor component, populates the average strain or stress tensor, and creates a study with a load case for each load group. When the calculation of an average material property is requested, the computed values are by default presented in a separate evaluation group under **Results**. If no material property evaluation is requested, or if it is not available, the results can be used to study the response of the RVE to the applied load cases.

You can also create a global material using the **Create Material** option under the **Study and Material Generation** button. This is available for the **Average strain** or **Average stress** periodic conditions. This option creates a material that contains the homogenized elasticity or compliance matrix, which can be accessed by other components in the model. Note that if you want to use the computed material properties in another model or store them in a user-defined material library, use the **Create Material by Value**

option. For the **Create Material by Reference** option, the automatically generated global material contains variables linked to the **Cell Periodicity** node in the current model.

	Micromechanical Model of a Fiber Composite: Application Library path **Structural_Mechanics_Module/Material_Models/ micromechanical_model_of_a_fiber_composite**

Modeling Bolts, Fasteners, and Welds

In this section:

- Modeling Pretensioned Bolts
- Simplified Modeling of Bolt Threads
- Simplified Modeling of Fasteners
- Modeling Spot Welds
- Evaluating Stresses in Continuous Welds

Modeling Pretensioned Bolts

The information about pretensioned bolts is applicable if your license includes the Structural Mechanics Module.

Bolted joints are common in mechanical and civil engineering structures. If you are interested in analyzing the details of a bolted joint, the prestress in the bolt must be taken into account in order to correctly capture the behavior under service loads. The Bolt Pretension functionality in COMSOL Multiphysics is designed to simplify such analyses. You can model the bolts either using solid or beam elements.

During mounting, a bolt is tightened to a certain prestress. The mounting of the bolt is, in general, accompanied by deformations of the surrounding structure. In the subsequent service, the force in the bolt can then change due to external loads.

Sometimes the sequence in which the bolts are tightened is important. This will be the case if significant nonlinearities are induced by the tightening process. Modeling such a process is also fairly straightforward. This is, however, a less common case which adds some extra complexity to the modeling, so it will be discussed separately below.

MODELING THE BOLTS

You must use a specific modeling technique in order to use a bolt in a prestress analysis.

Using Solid Elements

- You can model bolts using solid elements in 3D or 2D axial symmetry. In 3D, it is usually most efficient to add the predefined bolt geometries from the Part Libraries.

In 2D axial symmetry, the bolt is always assumed to be axially symmetric, and thus having the Z-axis as its center of rotation.

2. Make sure that there is at least one interior boundary perpendicular to the bolt axis somewhere in the shank. In the following, this boundary is referred to as the *slit boundary* (Figure 2-34). The slit boundary can be composed of several adjacent boundaries in the geometry.
3. If you are using bolts from the Part Libraries, a slit boundary is predefined, and has the selection name `Pretension cut`. In order to make this boundary selection visible from the physics interface, select its **Keep** checkbox in the **Boundary Selections** section of the settings for the part instance (Figure 2-35).
4. If needed, add contact conditions between the bolt head and the component and between different components clamped by the bolt. In many cases, it is sufficient and more efficient to use a continuity condition between the bolt head and the component.
5. Add a **Bolt Pretension** node, in which the pretension force or stress is prescribed for a set of bolts sharing the same data.
6. For each bolt having the same essential data, add a **Bolt Selection** subnode where its slit boundary is selected.
7. The label of the bolt, which is an input in the **Bolt Label** section, is used to identify the bolts during result evaluation. A suggestion for the name is automatically generated, based on the base name given in the parent node.

When a bolt is located in a symmetry plane (so that only half the bolt is modeled), and **Automatic symmetry detection** is selected in the **Bolt Selection** node, the given pretension force is interpreted as the force in the whole bolt, and not as the force in the half bolt. This makes it possible to use the same **Bolt Pretension** node for a set of similar bolts where some of them are located in symmetry planes.

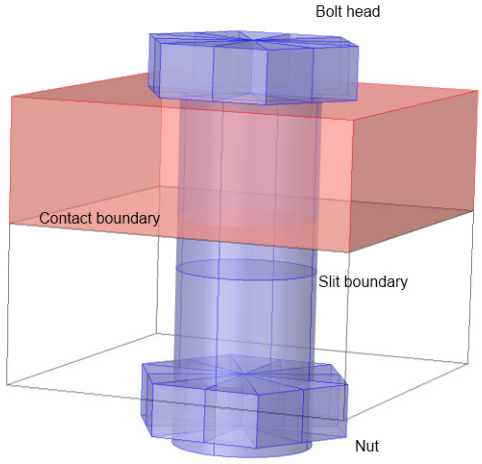


Figure 2-34: Example of a bolted joint with the bolt modeled as a solid.

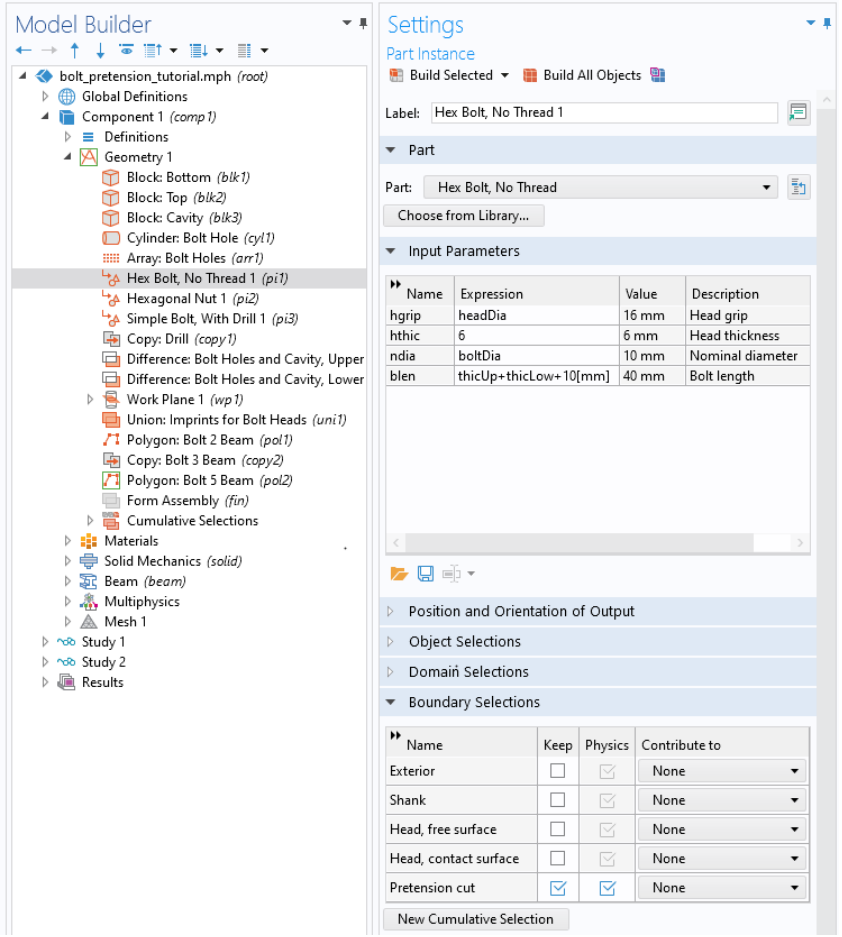


Figure 2-35: Getting access to the slit boundary selection for a bolt from the Part Libraries.

Using Beam Elements

1 When modeling with beam elements, you will typically use a **Polygon** with three points to model each bolt (Figure 2-36).

2 There must be at least one interior point somewhere in the shank. In the following, this point will, in analogy with the solid, be referred to as the s*lit boundary*.

3 Since there is no explicit bolt head when you model a bolt using beams, you need to connect it to the solid component using some type of abstract coupling. There are several options:

- You can connect the end of the beam to the edge of the bolt hole (the lower end in Figure 2-36). Then, use the Solid–Beam Connection multiphysics coupling, with **Connection type** set to **Solid Edges to Beam Points**. In this case, you select the edge of the bolt hole.
- You can create a circular boundary having the diameter of the bolt head on the surface of the component (the upper end in Figure 2-36). Then, use the Solid–Beam Connection multiphysics coupling, with **Connection type** set to **Solid Boundaries to Beam Points**. In this case, you select the annular solid boundary.
- Even without creating an extra boundary, you can use the Solid–Beam Connection multiphysics coupling, with **Connection type** set to **Solid Boundaries to Beam Points, general**. Then set **Connected region** to **Distance (manual)** and enter half the head diameter as **Connection radius**. In this case, you select the entire boundary on which the bolt head is residing. This method is convenient, but will often give more spurious stresses around the bolt hole, since partial element faces will be connected.
- You can use the two first approaches, but rather than using a multiphysics coupling for the connection, you can add rigid connectors in both physics interfaces, and then connect them. This approach is more expensive, since it adds degrees of freedom in both the rigid connectors, plus the constraints to couple them. It is however an approach that may be suitable for imported meshes, where connection data in this form is available.

All these techniques are shown in the Structural Mechanics Module Application Library example *Modeling of Pretensioned Bolts*.

4 Add a **Bolt Pretension** node, in which the pretension force or stress is prescribed for a set of bolts with the same data.

5 For each bolt having the same essential data, add a **Bolt Selection** subnode where its slit boundary is selected.

6 The label of the bolt, which is an input in the **Bolt Label** section, is used to identify the bolts during result evaluation. A suggestion for the name is automatically generated, based on the base name given in the parent node.

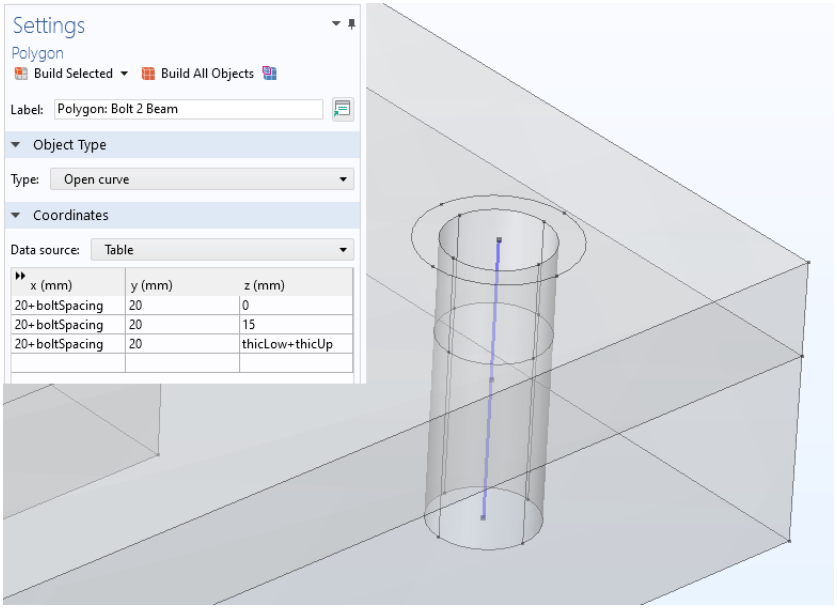


Figure 2-36: A bolt modeled using the Beam interface.

THE PREDEFORMATION DEGREE OF FREEDOM

Each bolt defined in the **Bolt Selection** node has a single global degree of freedom called predeformation, d. At the slit boundary, the two sides of the bolt are disconnected so that the displacements over it can be discontinuous. The discontinuity is represented by:

$$\mathbf{u}_{\mathrm{u}} = \mathbf{u}_{\mathrm{d}} - (d - r)\mathbf{n}$$

Here the subscript `u` denotes the upside of the slit boundary, and `d` denotes the downside. $\mathbf{n}$ is the normal pointing out from the downside. The sign has been selected so that d gets a positive value when the bolt force is tensile. An optional relaxation r can also be included.

The axial force in the bolt is thus caused by a small overlap between the two sides of the slit boundary. It is computed as the reaction force belonging to the degree of freedom d.

It is only meaningful to introduce the relaxation in a later study step. If it is present all the time, then its only effect would be to increase the predeformation during the pretension analysis by r. Thus, the value of r is usually a function of the load history, which is initially zero.

SETTING UP THE STUDY STEPS

In an analysis of prestressed bolts, you have to use two or more separate study steps. They can be part of a single study or be placed in different studies. The first study step, in which the bolt prestress is prescribed, simulates the mounting process. If you would only use a force to load the bolt (for example, as an initial stress), the resulting stress in the bolt would be less than the intended, due to the compression of the material around the bolt. The prestress step ensures that the bolts have the intended prestress, irrespective of the flexibility of the surrounding structure and their interaction.

In the subsequent studies, the bolt force is allowed to change, while keeping the extension of the bolt, as caused by the first study, fixed. The procedure to do this is as follows:

1. Run the study step for the mounting simulation. The predefined study type **Bolt Pretension** is designed for this. You may need to apply the pretension load in smaller steps, if there are nonlinearities in the system. Then, you select **Auxiliary sweep** in the **Study Extensions** section in the settings for the **Bolt Pretension** study step. Introduce a load ramping parameter, which is used to multiply the pretension forces.
2. Add one or more studies or study steps to analyze the effects of the service loads.
3. Since the pretension degrees of freedom are not solved for in the service load study steps, they must obtain their values from the pretension study step. If the study steps are sequential within the same study, no special action is needed, since the default then is to inherit values from the previous study step. For other cases, go to the **Values of Dependent Variables** section of the study step, set **Values of variables not solved for** to **User controlled**, and then select the pretension study step.

SEQUENTIAL TIGHTENING

If you need to take the order of the bolt tightening into account, then you must use an auxiliary sweep where the sweep parameter is used to control the tightening history.

You need to perform the following steps:

1. In each **Bolt Selection** node representing a bolt that is not fully pretensioned from the beginning of the study step, select the **Sequential tightening** checkbox.

2. A new text field, **Pretensioning expression**, is now shown. In this text field, you enter a Boolean expression, which evaluates to a nonzero value at the parameter values when the prestress is applied. It may happen more than once, if the bolt is not given its full pretension force at once. An example of such an expression is `round(par)==3 || round(par)==11`.
3. If you want the bolt prestress (when applied) to have a value that differs from the value given in the parent **Bolt Pretension** node, change the setting of **Pretension type** from **From parent** to another option. Extending the example above, the expression for the pretension force could be `50[kN]*if(par>3.5, 1, 0.6)`. The only thing that matters here is the value of the force at the parameter values when the pretension force is set, in this example 3 and 11.

A sequential tightening process may have a quite slow convergence rate until a certain number of bolts have been loaded. The reason is that it is common that two mating boundaries barely touch, so that many contact iterations are needed. It is advisable to

start by giving all bolts a small prestress, maybe 1% of the final value, in order to stabilize the model.

It is important to make sure that you only solve for the bolt predeformation degrees of freedom in the pretension study step, and not when analyzing the service loads.

If you use the **Bolt Pretension** study type for the pretension study step, and any other study type to analyze the service loads, the solvers are automatically set up to handle this. The **Bolt Pretension** study type is actually a special case of a **Stationary** study step, with the sole purpose of activating the predeformation degrees of freedom. These degrees of freedom are by default not solved for in any other study type.

If you, however, set up your studies manually, the information below is useful.

You enable or disable the solution of individual degrees of freedom under the **Dependent Variables** node for a certain study step in the solver sequence. If required, begin by clicking **Show Default Solver** in the study node or in the **Solver Configurations** node of the study. Then move to the **Dependent Variables** node, and in the **General** section, set **Defined by study step** to **User defined**.

You can now go to the node for each predeformation degree of freedom below **Dependent Variables** and adjust the state of the **Solve for this state** checkbox.

For more information, see also Dependent Variables and Studies and Solvers in the *COMSOL Multiphysics Reference Manual*.

RESULTS

The results in a bolt do not belong to any part of the geometry, but are global variables. To access the result from a certain bolt, a full scope of the type `<interface>.<Bolt Pretension tag>.<Bolt Selection tag>.<variable>` must be used. An

example could be `solid.pblt1.sblt1.F_bolt`. The bolt results are summarized in the table below.

TABLE 2-17: BOLT VARIABLES.

VARIABLE	DESCRIPTION
`d_pre`	Predeformation
`F_bolt`	Axial force in the bolt
`F_shear`	Shear force in the bolt (3D only)

If you place a bolt in a symmetry plane, that only half of the bolt is modeled, this will automatically be detected. The results are reported for the whole bolt, not for the symmetric half.

When there are pretensioned bolts in a study, evaluation groups containing the bolt forces will automatically be generated.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

- *Modeling of Pretensioned Bolts*: Application Library path **Structural_Mechanics_Module/Tutorials/bolt_pretension_tutorial**
- *Prestressed Bolts in a Tube Connection*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/tube_connection**

Simplified Modeling of Bolt Threads

When an internal and an external thread are engaging, there will be a complex local stress state. When analyzing such geometries, you can approach the problem at three levels:

- Make a full 3D model of the actual thread geometry and solve the contact problem. This approach is only used in some rare cases, since such a model by necessity becomes very large. Also, the actual stress state in the thread itself is seldom the target of an analysis.

- Ignore the thread completely and just connect the two parts by a union operation or through a **Continuity** pair. This approach will provide an accurate solution outside a region with a size of a couple of bolt diameters from the bolt hole.
- Use the simplified **Bolt Thread Contact** condition. With this approach, the stresses will be correct, except at distances where stress concentrations caused by the actual thread geometry are significant. The important part of the solution that can be captured by this simplified contact condition is that the contact pressure between the threads will push the walls of the bolt hole outward. This will cause significant tensile hoop stresses around the bolt hole.

When you use **Bolt Thread Contact**, you model the face of both the bolt and the bolt hole as cylinders. The actual geometry of the thread is taken care of by the mathematical formulation of the contact condition. The most important parameter is the thread angle, since it determines the direction of the contact forces.

For each individual bolt, you need to add one **Bolt Thread Contact** node.

PENALTY FACTOR

The contact condition is enforced using a penalty formulation. In practice, this means that a stiff spring is inserted between the two boundaries. In most contact problems, you want to use a high penalty factor, in order to avoid excessive overclosure of the contacting boundaries. In this case, however, there is a certain flexibility which has its source in a slight bending of the threads in real life. This effect can to some extent be accounted for by allowing a certain flexibility in the contact condition. For this reason, the default penalty factor is rather low. You may need to adjust it to suit your conditions.

A higher penalty factor also means that the force is transferred from the external thread to the internal thread over a shorter distance along the bolt axis.

CONTACT ORIENTATION

In most cases, you know a priori which side of the thread that will be in contact. You can reduce the size of the contact problem by a factor of two by selecting an appropriate value for the **Contact orientation**. The default is to search in both directions.

If a bolt is pretensioned, the contact orientation will be **Up** if the bolt orientation is such that the **Orientation** vector points toward the bolt head. If the bolt geometry, including orientation, is computed using the **Automatic** option, you will often have to use the **Direction adjustment** option to have full control over the up and down directions.

MODELING HINTS

- You must create one **Contact Pair** for each bolt and mating bolt hole. The bolt should be the destination side of the contact pair.
- In the Part Libraries, you will find predefined parameterized bolt geometries that also contain a domain intended to be used as drill for the bolt hole.
- The cylindrical surfaces of the bolt and bolt hole should have the same diameter. If you want to decrease the stiffness of the bolt because it is not solid all the way out to the nominal diameter, a good suggestion is to decrease Young's modulus by a factor based on the *stress area* of the bolt.
- Since the contacting boundaries are cylindrical, there is a risk that the bolt, instead of computing the gap to the nearest point on the bolt hole, will instead see a point on the opposite side. A good practice is to select **Manual** from the **Search distance** list in the settings for the **Contact Pair**. Half the diameter of the bolt is an appropriate search distance.
- Since the relative displacements between the bolt and the hole are small, setting **Mapping method** to **Initial Configuration** in the settings for the **Contact Pair** can improve efficiency.
- In the initial state, the two objects being joined may possess rigid body modes, just like in any other contact problem. If so, you can for example add weak springs to maintain stability.

Simplified Modeling of Fasteners

APPLICABILITY

The Fasteners node is designed for the approximate modeling of large numbers of fasteners. Its primary use is for rivets, but with certain assumptions it can also be applied to bolts and similar connectors. For spot welds, a dedicated node is available and is described in Modeling Spot Welds.

Because the Shell interface does not represents stresses in the direction normal to the shell, any prestress in the fasteners is ignored. In contrast, the Solid Mechanics interface allows the inclusion of a preload in the fastener analysis.

When modeling pretensioned fasteners, it is important to include the contact between the connected plates, as most of the load is transmitted through the joint via contact pressure or friction force. The total load is the sum of the load carried by the bolt and the load carried through the joint. As the applied load increases, the bolt load increases

while the contact forces decrease, until the plates separate. At this point, the entire load is carried by the bolt, which is then considered as a loose bolt.

In the absence of preload, the forces are transmitted solely through tension and shear in the fastener. This is a common assumption when modeling rivets. When using the Shell interface to model pretensioned bolts, this approach leads to the following consequences:

- The axial force or stress in the bolt caused by the tensioning of the bolt is ignored.
- Any axial force or stress caused by external loads is strongly exaggerated, since in a proper bolt connection, a large portion of a load that is superimposed on the prestress is carried by the surrounding material.
- Tangential forces in a well-designed bolted connection are carried by friction between the joined objects. Thus, the shear force reported by the fastener model is not representative for the shear force in a bolt.

These limitations are, to a large extent, related to the local stress evaluation in the bolt. However, there is also a certain underestimation of the stiffness. This can, to some extent, be mediated by increasing the fastener stiffness.

MODELING USING THE FASTENERS NODE

A fundamental assumption is that the fastener hole is present in the geometry of both plates to be joined. The actual connection occurs between two sets of hole edges, or boundaries. As long as the pairs of holes are coincident within some tolerance, all fastener locations can be found automatically. This is the standard way in which you set up a model using the **Fasteners** node.

A **Fasteners** node has one or more Hole Selection subnodes. The automatic selection mode is invoked by the choosing **Automatic** in the **Fastener Hole Detection** section in such a subnode.

For two holes to be automatically connected by a fastener, the following criteria must be met:

- One of the holes belongs to the set of entities selected as source, and the other one belongs to the set of entities selected as destination.
- The hole diameter cannot be smaller than the fastener diameter, but it can be larger within a certain tolerance (**Diameter tolerance**).
- The centers of the two holes should be located on a common line parallel to the normal on the source side. You can adjust the tolerance (**Offset tolerance**).

- The outer boundaries should be parallel, and the direction of their normals should not differ more than a small angle. You can adjust the maximum angle difference (**Maximum angle**).
- When the selection entity is set to boundary, select **Search annular shape only** to detect holes that have surrounding boundaries with annular shapes. The center of the inner and outer circles may differ within a certain tolerance (**Annular center offset**).

In Figure 2-37, an automatic fastener connection is illustrated. In the upper (gray) plate there are 16 holes. Nine of them have a matching hole in the lower (yellow) plate. One of these holes, however, have a larger diameter. With the default tolerances, eight fasteners will be generated. In the lower part of Figure 2-37, the result template **Connected Region Indicator** is shown. This plot is an important tool for checking several types of connections.

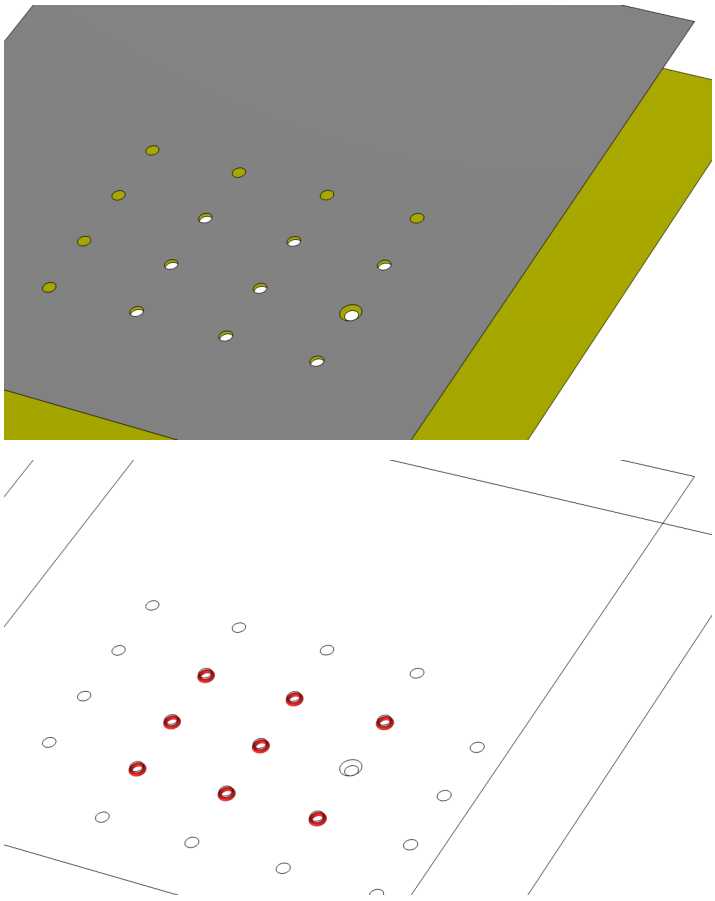


Figure 2-37: Example of automatic detection of fasteners.

If the conditions for hole matching above are not fulfilled (or cannot be fulfilled by reasonable changes of the tolerance values), you will need to make manual selections for the holes. In that case, you need one **Hole Selection** subnode for each fastener. Change the setting to **Manual** in the **Fastener Hole Detection** section, and make the appropriate entity selections. By using **Group by Continuous Tangent**, it is easy to select all edges around a hole. In this mode, it is not possible to create more than one fastener per **Hole Selection** subnode.

You can mix automatic and manual selections under the same **Fasteners** node.

When analyzing riveted structures, it may be necessary to also model contact between connected parallel plates in order to avoid that they penetrate each other. In most cases, a set of fasteners will constrain the plates to such an extent that the advantage of actually including contact in the model is small.

When analyzing pretensioned fasteners using the Solid Mechanics interface, it is necessary to model contact between the connected plates.

In the Shell interface, the sides of the shells (top or bottom) that are to be connected must be known. The information is used for two purposes: to connect the correct set of displacements, and to compute the distance, including the shell thickness and offset. With the default settings, **Automatic**, the two sides of the shells that are found to be closest to each other are connected. However, such an automatic search can take a significant time if many boundaries are coupled, or there are many mesh elements. If this is a problem, you can:

- Split the selections into several **Fasteners** nodes.
- In the **Advanced** section in the settings for **Fasteners**, change the **Search method** to **Manual**, and set a suitable **Search distance**.

RESULTS

In many cases, you are only interested in the fasteners as a means of obtaining a proper global stiffness of a structure. You can, however access the results in the individual fasteners. There are two result quantities: the axial force and the shear force.

From the **Result Templates** window, you can add

- A **Fasteners Forces** plot group, to visualize the normal and shear forces at each fastener, and a label showing each fastener number. Normal forces are plotted as green arrows, and shear forces as blue arrows. The coloring can be affected by the addition of a **Safety** subnode. To display the fasteners forces, these have to be evaluated first using the **Fasteners Normal Forces** and **Fasteners Shear Forces** evaluation group.
- A **Fasteners Normal Forces** evaluation group that contains a table of the location, the normal forces, and the damage index of all fasteners. This table has to be evaluated to display the forces in the **Fasteners Forces** plot group.

- A **Fasteners Shear Forces** evaluation group that contains a table of the location, the shear forces, and the damage index of all fasteners. This table has to be evaluated to display the forces in the **Fasteners Forces** plot group.
- A **Fasteners Forces Magnitude** evaluation group that contains a table of the normal and shear forces magnitude of all fasteners. In the table, click **Show Row Numbers** to display the number of each fastener.

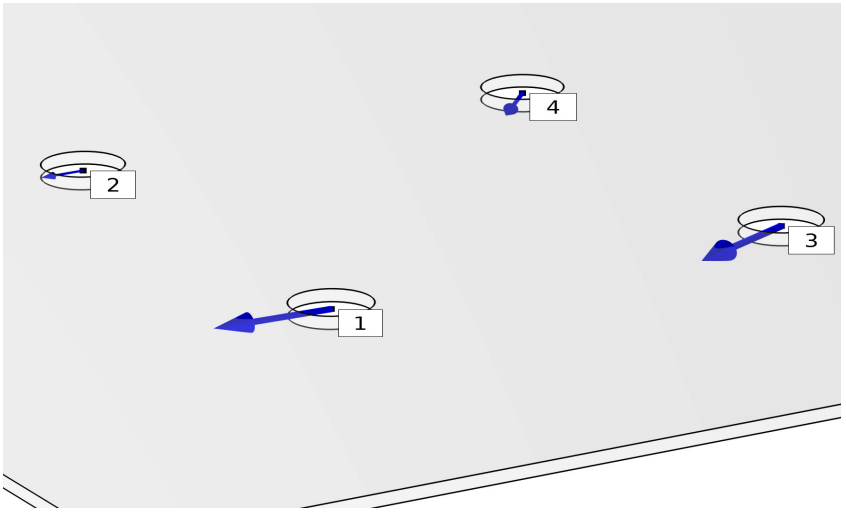


Figure 2-38: Example of a results from a Fastener Forces plot group.

By adding a **Safety** subnode under **Fasteners**, you can evaluate the degree of utilization for each fastener.

The damage criterion is

$$\left(\frac{F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} = 1$$

Here, $F_{\mathrm{n,max}}$ and $F_{\mathrm{s,max}}$ are the critical forces in tension and shear, respectively. The exponents α_{n} and α_{s} are often chosen as 2 for rivets, while other choices are common for bolts, for example $\alpha_{\mathrm{n}} = 2$ and $\alpha_{\mathrm{s}} = 3$.

If the allowable forces are exceeded, that is

$$\left(\frac{F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} > 1$$

then the forces in the **Fasteners Forces** plot group will be colored in red.

You can manually specify a **Safety factor** sf_0 to evaluate the intermediate degree of utilization before failure, that is

$$\left(\frac{\mathrm{sf}_0 F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{\mathrm{sf}_0 F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} > 1$$

In this case, the forces in the **Fasteners Forces** plot group will be colored in orange.

You can also specify a **Margin of safety** ms_0, which is related to the safety factor as

$$\mathrm{sf}_0 = 1 + \mathrm{ms}_0$$

Modeling Spot Welds

INTRODUCTION

The Spot Welds node in the Shell interface is intended for global analysis of structures connected with a large number of spot welds. The details of the spot weld as such is not modeled. An elastic connection is established between the two shell faces.

The connection is mesh independent, and the locations of the spot welds are specified using a list of coordinates. However, you cannot use arbitrarily large elements. The element size at the spot weld location should not be significantly larger that the diameter of the weld nugget.

MODELING USING THE SPOT WELDS NODE

The location of the welds is given in a table in the **Nugget Location** section. In most cases, you will import this table from an separate text file using **Load from file** ( ).

Nugget File Format

The format of the file is simple. Each weld is represented by a single line with 4 entries separated by white space:

```
<Nugget_number> <X_coordinate> <Y_coordinate> <Z_coordinate>
```

Any line that starts with a hash sign (#) is treated as comment and is ignored.

The exact location of the nugget does not have to be perfect. The given location is projected to the nearest point on the destination boundary. Thus, the nugget coordinates can, for example, represent an average location of the nugget (between the plates) or be placed on either plate.

Connected Location

The coupling is intended to operate between two adjacent physical surfaces, possibly at some small distance. That is, for each of the connected shells, it is either the top or the bottom surface that is connected. In most cases, the surfaces to be connected can be determined automatically. The algorithm measures the four possible distances between top and bottom sides of source and destination, and use the pair that has the smallest distance.

You can choose to manually specify which surfaces to connect. To do that, change **Connected location** from **Automatic** to **Manual**.

Nugget Stiffness

The elastic properties of the nugget is represented by an axial stiffness (k_n) and a shear stiffness (k_s). You can provide these values explicitly.

The default is to compute the stiffness using the properties of a thin layer. The thickness of the layer is a user input, named **Cohesive thickness**. The stiffness of the weld is inversely proportional to this value. Physically, the cohesive thickness can be seen as representing the part of the nugget that is located between the two plates. This is, however, not the full truth, since the local stress and strain field around a real spot weld is complex. Even an ideal weld, where there is no distance between the plates, will have some flexibility.

When computing the nugget stiffness, the elastic properties are taken from the destination boundary.

Mesh Size and Numerical Integration

The spot weld connection is mesh independent, in the sense that it is neither a node-to-node connection or a specific mesh element. Instead, a thin elastic layer is added over an area that approximates the size of the weld nugget.

Still, to achieve good accuracy the mesh must not be too coarse; the coupling uses a numerical integration on the destination boundary, and a large enough number of integration points are needed. A rule of thumb is that the element size (in terms of edge length) should be smaller than the nugget diameter.

More information about numerical integration can be found in the following blog post:

www.comsol.com/blogs/introduction-to-numerical-integration-and-gauss-points

When a coarse mesh is used, you can improve the accuracy by increasing the number of integration points per elements. To do that, clear the **User automatic quadrature settings** checkbox in the **Quadrature Settings** section.

Figure 2-39 shows conceptually how two boundaries are connected by a spot weld.

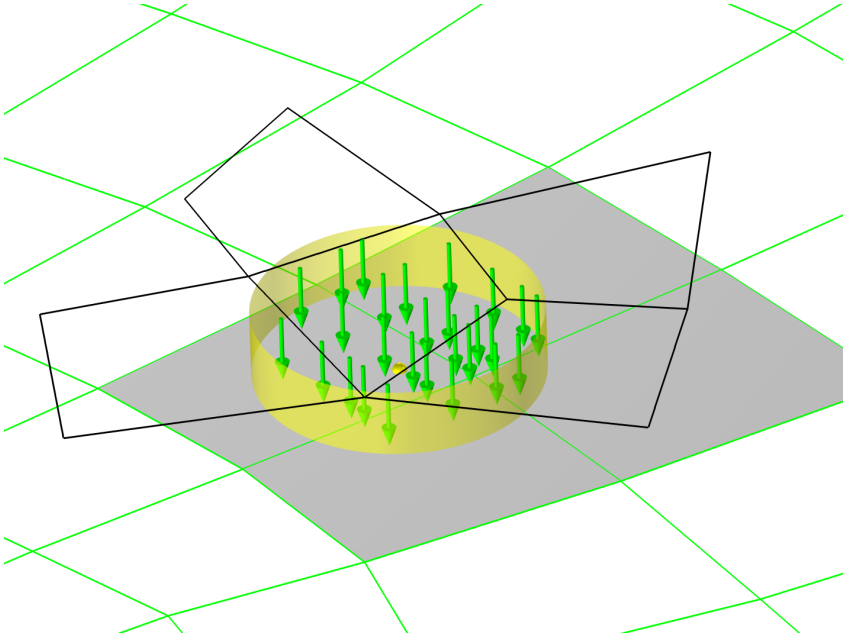


Figure 2-39: Illustration of spot weld connection.

Below is a detailed explanation of the figure:

- The yellow ball represents the nugget location on the destination boundary. It may have been placed there by projection.
- The yellow cylinder represents the nugget diameter. It has the nugget location at its center.
- The green mesh is located on the destination boundary.
- The gray elements are the elements on the destination boundary that are involved in the connection. They have at least one integration point (Gauss point) inside the nugget diameter.
- The black elements are the elements on the source boundary that are involved in the coupling.

- The green arrows indicate how displacements are mapped from the source boundary to destination boundary.
- Each arrow has its head located at an integration point on the destination boundary. The default integration scheme, shown here, uses 4x4 integration points per element.

You can check the interpretation of the connection by adding a **Nugget Location** plot from **Result Templates**. as shown in Figure 2-40. In this plot, each nugget is indicated using a red color. Also, each integration point that contributes to the nugget stiffness and force calculation is shown in blue.

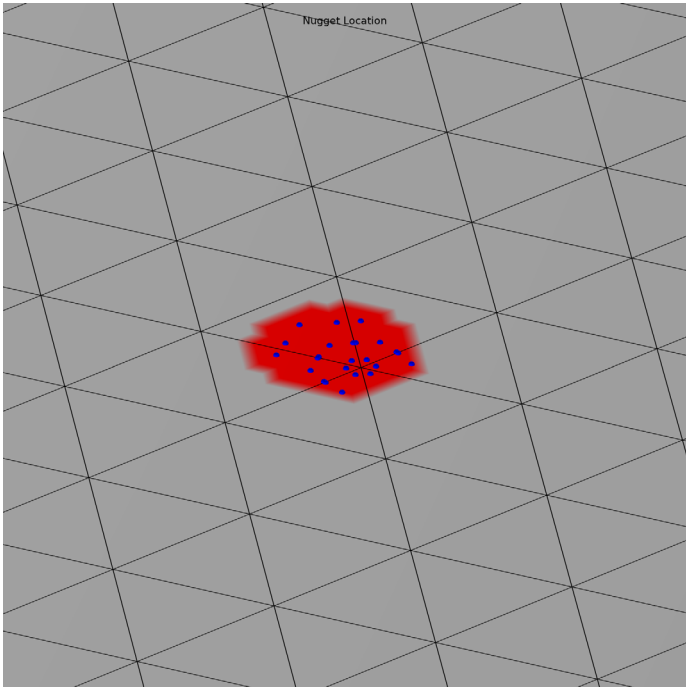


Figure 2-40: Closeup in a Nugget Location Plot.

If you want to investigate the distribution of integration points using another integration scheme, you can modify the **Gauss-point order** setting in the **Evaluation** section in the settings for the **Point 1** plot in the **Nugget Location** plot group.

Performance Considerations

The mapping between the connected boundaries can take a significant time if many boundaries are coupled, or there are many mesh elements. If this is a problem, there are three methods you can use improve the situation:

- Split the selections into several **Spot Welds** nodes.
- Set **Connected location** to **Manual**.
- In the **Advanced** section in the settings for **Spot Welds**, change the **Search method** to **Manual**, and set a suitable **Search distance**.

RESULTS

In many cases, you are only interested in the spot welds as a means of obtaining a proper global stiffness of a structure. You can also access the results in the individual welds.

From the **Result Templates** window, you can add:

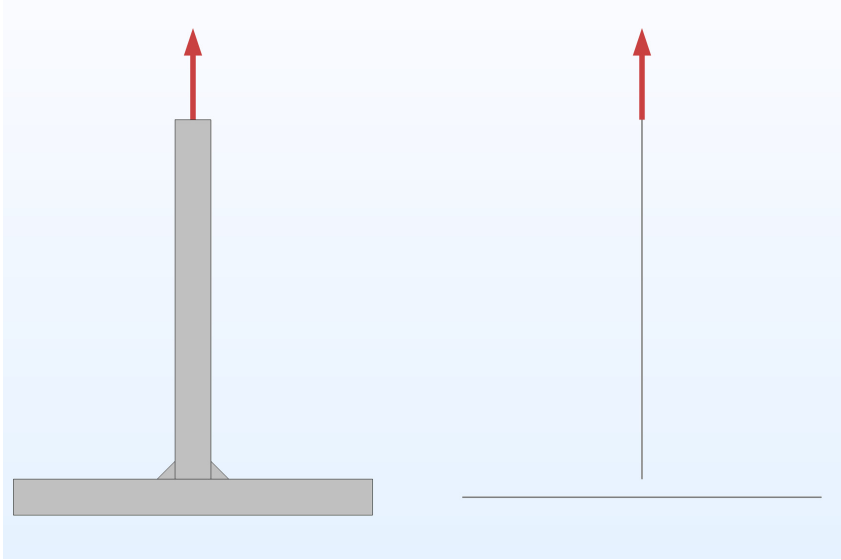
- A **Spot Welds Forces** plot group, to show the normal and shear forces at each weld, and a label showing the identity number of each weld. Normal forces are plotted as green arrows, and shear forces as blue arrows. To display the spot weld forces, these have to be evaluated first using the **Spot Welds Normal Forces** and the **Spot Welds Shear Forces** evaluation groups.
- A **Spot Welds Normal Forces** evaluation group that contains a table of the location and the normal forces of all welds. Evaluate this table to display the forces in the **Spot Welds Forces** plot group.
- A **Spot Welds Shear Forces** evaluation group that contains a table of the location and the shear forces of all welds. Evaluate this table to display the forces in the **Spot Welds Forces** plot group.
- A **Spot Welds Forces Magnitude** evaluation group that contains a table of the normal and shear forces magnitude of all welds. In the table, select **Show Row Numbers** to display the number of each fastener.

Other results that are available are the bending and twisting moment in each spot weld, as well as the total force transferred by all spot welds in a **Spot Welds** node.

Evaluating Stresses in Continuous Welds

Weld evaluation consists on the calculation of stresses and forces transmitted by the weld. These are computed on the plane of the weld throat, and different criteria can be used to compute equivalent stresses and safety indexes. It is currently possible to

perform weld verification in the **Shell** interface for double-sided and single-sided fillet as well as butt joints. It is assumed that the 3D solid model can be approximated by shells.



To perform a weld evaluation, you select the **Weld verification** checkbox in the **Edge to Edge** feature in the Shell interface.

The weld evaluation is a postprocessing tool and it is performed during result presentation. It is not necessary to recompute the solution if the welding type, the weld size, or the allowable stresses are changed. It is enough to update the solution.

The evaluation is based on the shell section forces at the weld edge on the source side of the connection. From these forces, available from the **Shell** interface, only the components that are transmitted through the weld are kept. The forces are transformed to the weld reference frame. The equivalent stress is computed as follows:

$$\sigma_{eq} = \sqrt{\sigma_{perp}^2 + 3(\tau_{perp}^2 + \tau_{para}^2)}$$

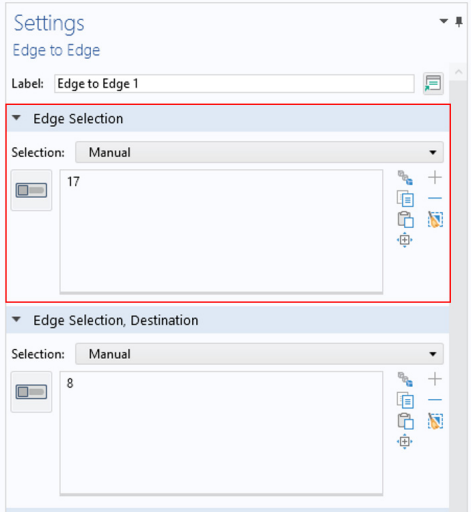
where the contribution of the normal stress parallel to the weld throat is neglected because it does not transfer any load through the weld.

For a single-sided fillet, the top and bottom options switch the sign of the eccentricity used to compute the additional stresses due to bending.

Different safety indexes are available to evaluate the weld: failure index, damage index, safety factor, and margin of safety.

SOURCE AND DESTINATION SELECTION

When interested in weld evaluation the source and destination in the **Edge to Edge** feature needs to be selected in a proper way to obtain meaningful results for the one-sided and double-sided fillet joints. The source has to be the edge belonging to the plate that would have both top and bottom sides welded if a double-sided fillet was to be used.



LIMITATIONS

The weld evaluation for the fillet welds assumes that the connecting shells are perpendicular to each other and the angle between the shell and the throat plane is 45°.

When using a double-sided fillet weld, one side is more critical than the other. The results will always show the safety factors on the most critical side, where usually the shear stresses are greater. The single-sided fillet is treated in the same manner.

VARIABLE NAMES

When the weld evaluation checkbox is selected, a set of new variables are defined. All of them are item scoped and the most important are listed below.

TABLE 2-18: VARIABLES FOR WELD EVALUATION.

VARIABLE	DESCRIPTION
weld_di	Damage index
weld_ms	Margin of safety
weld_fi	Failure index
weld_sf	Safety factor
weld_seq	Equivalent stress along the edge
weld_tfwavgi	Traction vector average in the weld reference system, i component
weld_tfeavgi	Traction vector average in the edge reference system, i component

Modeling Thin Layers

Sometimes, there are thin layers between larger components that cannot be ignored in a structural analysis. The reasons are either because they have significant flexibility or damping properties, or that they may fail. Examples of such layers are glue layers, O-rings, and gaskets. Other cases can involve transition zones between particles or fibers and the binder in a composite material.

When the layer is thin compared to other dimensions of the structure, it is difficult to produce a mesh with acceptable quality without compromising the numerical conditioning of the stiffness matrix. The **Thin Layer** boundary feature allows to overcome this problem by making simplifications, so that there is no need for a mesh in the through-thickness direction.

There are three different approximations available for modeling thin layers:

- Solid
- Membrane
- Spring

These approximations differ in the simplifications made for obtaining the deformation gradient in the thin layer. For more details see Thin Layers.

The Thin Layer Node

The **Thin Layer** node is used to identify boundaries that have a thin layer attached to them. These boundaries may either be exterior or interior boundaries to the domain where the physics interface is active. The **Thin Layer** node also determines the thickness of the thin layer, and how to approximate the deformation gradient on it. The first choice is whether to use a **Layered** or a **Nonlayered** thin structure.

In the majority of cases, the **Nonlayered** approximation is sufficient to capture the behavior of the thin layer. For this type of layer, two sets of degrees of freedom (DOFs) are introduced at the boundary; one set corresponds to the upside, and one set to the downside as illustrated in Figure 2-41. In COMSOL Multiphysics, this is referred as a slit of the displacement field on the thin layer boundary.

Note that in Figure 2-41 no extra integration points are introduced in the through-thickness direction, and that numerical integration is made on the midsurface. Hence, dependent variables always have a linear variation in the thickness direction; that is,

their gradients are constant in the through-thickness direction. This also applies to inelastic quantities such as plastic strains.

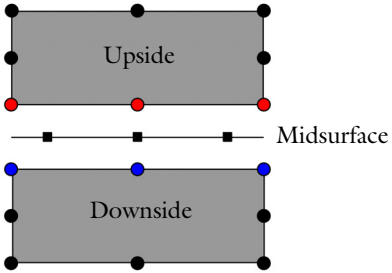


Figure 2-41: Illustration of a nonlayered thin stratum between two mesh elements using quadratic serendipity shape functions. Circles indicate nodal points and squares integration points.

The **Layered** type can be used when a more detailed description of the through-thickness behavior is needed. For this type of thin layer, a slit of the displacement field is made on the boundary, and additional DOFs are introduced between the upside and downside as illustrated in Figure 2-42. These extra DOFs facilitate a more detailed through-thickness representation of stresses and strains, and they make it possible to, for example, model composites and laminate strata. Continuity is enforced by constraints between the DOFs on the upside and the downside, and the extra DOFs. The through-thickness discretization is controlled by the attached material. For more details about modeling layered structures, see Composite Materials Modeling

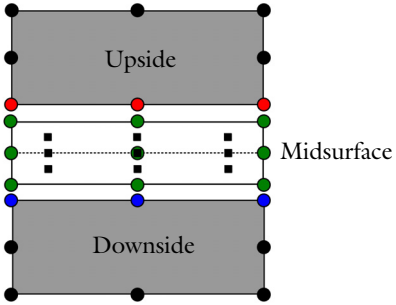


Figure 2-42: Illustration of a discretized layered thin layer using a single mesh element in the through-thickness direction. The thin layer is injected between two mesh elements using quadratic serendipity shape functions. Circles indicate nodal points and squares integration points.

When the **Nonlayered** type is used, it is possible to choose different approximations of the deformation measure in the thin layer:

- Solid
- Membrane
- Spring

The description in Figure 2-41 is representative of the solid and spring approximations. The solid approximation is the most general and accounts for the full normal and tangential deformations in the thin layer. The spring approximation is similar but neglects the membrane deformation. This is useful for cases where the behavior of the thin layer is given by force versus extension data, as is often the case when modeling gaskets.

For the membrane approximation, no slit is introduced and the thin layer only accounts for the membrane deformation. It thus only contributes with tangential stress, while plane stress conditions apply in the through-thickness direction. This is typically the case when the thin layer describes a cladding on an exterior boundary or a reinforcement layer on an interior boundary.

The different thin layer approximations are compared in *Thin Layer Interfaces*: Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/thin_layer_interfaces**

Interior and Exterior Boundaries

A **Thin Layer** can be applied to either exterior or interior boundaries of the physics interface. However, there are some noticeable differences between adding a thin layer on an exterior or on an interior boundary.

On an interior boundary, the selected boundary is always considered to be the midsurface of the thin layer. It is not possible to apply any loads or boundary conditions on interior boundaries.

On an exterior boundary, the bottom of the thin layer is attached to the adjacent domain. This means that the thickness of the thin layer extends outward from the domain in the direction of the normal. For a nonlayered thin structure, an exterior displacement field is introduced to create the slit, so it is possible to apply loads and boundary conditions to this exterior field. For example, applying a fixed constraint on an exterior thin layer makes it equivalent to a Spring Foundation. For a layered thin

structure, the *top side* of the layered material is always the exterior. Loads and boundary conditions can be applied to the top side only.

When the thin layer boundary is part of a contact pair, the thickness of the exterior thin layer is considered when modeling contact.

Material Models

A **Thin Layer** can be assigned different material models:

- Linear Elastic Material
- Nonlinear Elastic Material
- Hyperelastic Material
- Spring Material

The first three material models are equivalent to the corresponding domain material, hence the discussion in Introduction to Material Models is valid also for thin layers.

The **Spring Material** is a special material model only available with the spring approximation. As the name implies, data for this material model is given in terms of spring constants, or as force as a function of extension. It can be useful for certain types of modeling where data is given in this format, for instance, when normal force versus closure curves are available.

It is also possible incorporate inelastic effects by adding one or several subnodes to these material models.

Loads and Boundary Conditions

Several loads and boundary conditions can be applied to a thin layer. Exactly what boundary conditions are available and how they are treated depends on the thin layer approximation. Whether the thin layer is an exterior or interior boundary of the physics interface also affects the available boundary conditions.

On exterior boundaries, loads and boundary conditions can be applied to the exterior displacement field of the nonlayered stratum, or on the top side of a layered material for a layered stratum. For example, applying a fixed constraint to a thin layer on an exterior boundary makes is equivalent to a Spring Foundation. It is also possible to apply loads and boundary conditions to edges and points of the thin layer. For a nonlayered stratum, loads and other boundary conditions apply to the exterior

displacement field. However, for a layered stratum, all extra DOFs attached to the edge or point are constrained.

When a **Roller** is applied on a boundary perpendicular to a thin layer, a **Roller** should in most cases also be added to the edge of the thin layer. This is especially important if the roller represents a symmetry condition.

On interior boundaries, only loads and boundary conditions on edges and points are available. Moreover, these are only relevant for a layered stratum. For a nonlayered stratum, there are no free DOFs on which to apply the boundary conditions.

Result Evaluation and Visualization

For a nonlayered thin structure, many variables defined by the material models have the same name for both the thin layer and the adjacent domain. For example, the von Mises stress is always called `solid.mises`. Hence, the context in which the variable is evaluated is important.

If `solid.mises` is evaluated using a surface plot in a 3D model, it will have the following interpretation:

- On boundaries with no thin layer, it will show the von Mises stress of the adjacent domain.
- On boundaries that intersect a nonlayered thin layer, it will show the von Mises stress on the thin layer.

Use the `mean()` and `side()` operators to force the evaluation of variables in adjacent domains. For example, using `mean(solid.mises)` for a surface plot will average the stresses from the two adjacent domains, and it will not consider the stress in the thin layer between these two.

Extra care must be taken when evaluating results on entities of lower dimensions, such as points, to make sure that the expression is evaluated in the correct context. The `mean()` or `side()` operators can then be applied multiple times. For example, using `mean(mean(mean(solid.mises)))` in a point plot for a 3D model will ensure that the von Mises stress in the domain material is evaluated. Similarly, a point plot with the expression `mean(side(1,solid.mises))` will evaluate the von Mises stress in the thin layer on boundary 1.

	When one or more thin layers are present in a model, a default plot is added to show the stress in the thin layer.

	See Results Evaluation in Composite Materials Modeling for a discussion on how to evaluate and visualize results for layered thin layer.

Modeling Cracks

A crack in a solid model can be represented in two fundamentally different ways:

- As an ideal crack — that is, by a single boundary. Across this boundary, the displacement field is discontinuous. Using COMSOL Multiphysics nomenclature, this is called a *slit condition*.
- By the crack geometry, in which case the two sides of the crack are two different boundaries. In the true geometry, the two crack surfaces may coincide, or there may be a distance between them.

Both cases have an analogous representation in a symmetry plane. The ideal crack is just a boundary located in the symmetry plane, but without the symmetry conditions, so that it can open. This can, however, just as well be considered as special case of a geometrical crack, so in symmetry planes the distinction more or less disappears.

Fundamental for cracks is that the stress and strain states at the crack tip are singular, so any mesh refinement will only produce even higher stresses. For a linear elastic material, the stresses and strains in the vicinity of the crack vary as $1/\sqrt{r}$, where r is the distance from the crack tip.

The Crack Node

The Crack node serves several purposes:

- In the case of an ideal crack on an interior boundary, the degrees of freedom are split so that the displacements on the two crack surfaces are independent.
- In the case that a crack is located in a symmetry plane, all constraints that may be applied (such as symmetry) are removed. It is then possible for the crack to open.
- If possible, a crack growth direction is computed. For some cases this is instead a user input.
- A number of variables, such as an estimate of the crack length are created.
- A mesh suggestion is generated.
- By using the J-Integral subnode under **Crack**, you can compute J-integrals and stress intensity factors.

- By using the Virtual Crack Extension subnode under **Crack**, you can compute energy release rates and stress intensity factors.
- By using the Crack Closure subnode under **Crack**, you can add contact conditions between the crack faces.

The selected boundaries can form branched cracks.

MESH

At a crack tip, there will be an automatic mesh refinement. In the mesh sequence, you will find an extra **Size** node, with all crack tips selected. You may want to adjust the mesh size in that node.

J-Integrals

The J-integral is a path independent integral that can be used to characterize the severity of the stress state at the crack tip, both for linear and some nonlinear materials. You can compute J-integrals by adding one or more **J-Integral** subnodes under **Crack**.

Originally, the J-integral concept was derived for 2D, in which case it was sufficient to integrate certain functions of stress and strain along an arbitrary curve from one boundary of the crack to the other.

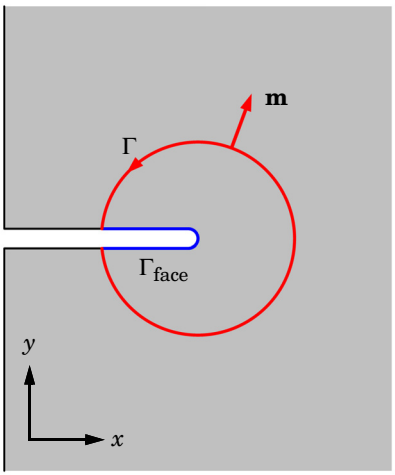


Figure 2-43: Example of a J-integral path.

There are some important assumptions when computing the J-integral:

- The curve is completely inside the domain.
- The curve does not enclose any other singularities, such as crack tips.
- No volume forces act inside the curve.
- No thermal stresses or similar inelastic strain contributions act inside the curve.
- If there is a load on the crack faces, this requires an extra correction. This is taken into account if the load is given using the **Face Load** subnode under **Crack**.

In cases where the J-integral is not applicable, you can instead use the Virtual Crack Extension Method.

You can specify the integration path as a circle with a given radius, in which case it cuts the mesh elements at arbitrary locations. In 2D, you can also select a sequence of interior or exterior boundaries describing the path. It is recommended that you evaluate more than one path to assess the accuracy of the solution.

The importance of this method is that stresses and strains far from the singular fields at the crack tip are used for the evaluation.

J-integrals can be computed also in 3D, but there are some complications. In the 3D case, the value of J will vary along the crack front. The local value is computed by a similar path integral, which then must be placed in the plane perpendicular to the crack front. This is automatically handled by the **J-Integral** node. The J-integral in 3D should be considered as a continuous function along the crack front.

In 3D, there is also a surface integral contribution which needs to be computed for the area that is enclosed by the curve. Unfortunately, this causes the singular field to enter the integration, a fact that reduces accuracy. In addition, the computational effort increases significantly. Evaluations of J-integrals in 3D can take a noticeable time.

Another problem, that exists only in 3D, occurs if the crack front terminates at a free boundary. If the crack front is not perfectly perpendicular to the free boundary, the integration path will not entirely remain inside the solid domain. This is manifested as spurious results close to the free boundary.

RESULT TEMPLATES

When **J-Integral** nodes are present, they will generate result templates. These plots reside in a **Cracks** plot group. The contents of the plots differ significantly between 2D and 3D, as described below.

2D

The integration paths are plotted as magenta curves. An arrow shows the crack growth direction. At the crack tips, the value from the last **J-Integral** node is printed. An example is shown in Figure 2-44.

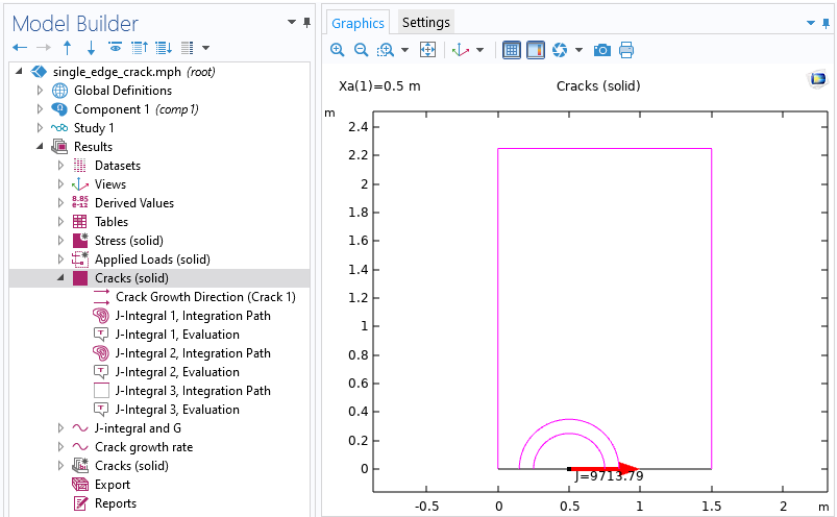


Figure 2-44: Result template for a 2D case with a crack in a symmetry plane. Three integration paths are shown; 2 circular and one along boundaries.

3D

In 3D, the integration paths are shown as transparent gray boundaries. The actual integration paths are located where this boundary is intersected by planes perpendicular to the crack front.

A colored arrow plot shows the crack growth direction, as well as the local value of the J-integral as computed in the last **J-Integral** node. An example is shown in Figure 2-45.

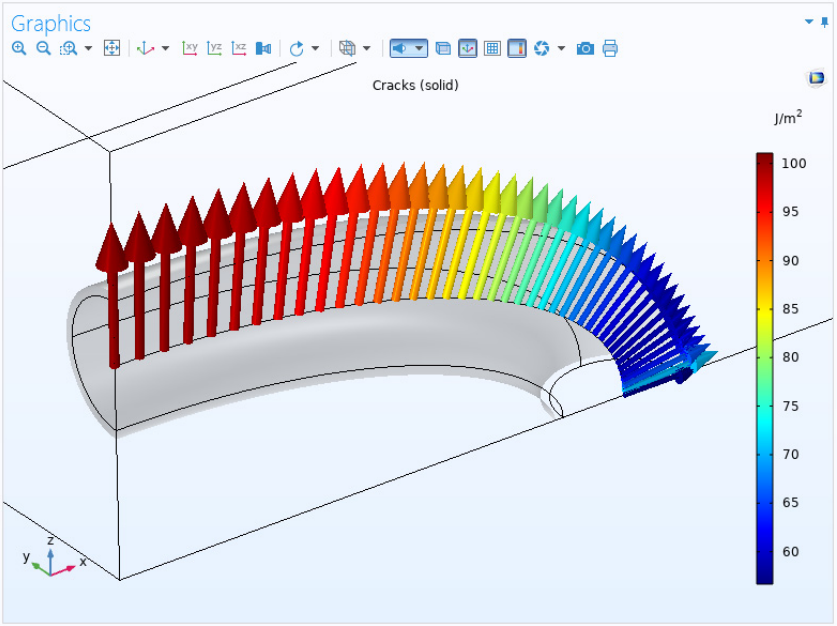


Figure 2-45: Result template for a case with a 3D crack in a symmetry plane. One integration path is used.

DEFAULT EVALUATION GROUPS

In both 3D and 2D, a predefined evaluation group, named **Fracture Mechanics Results**, is also generated. It contains a table with values of J-integrals and stress intensity factors for all integration paths. In 3D, however, only the maximum value of J along the crack front is reported.

Virtual Crack Extension Method

Similar to the J-integral method, the virtual crack extension method is used for computing the energy release rate and stress intensity factors. These two methods complement each other. The advantage with the virtual crack extension method is that it contains less assumptions. It can be used, for example, in the presence of body loads and thermal strains.

The virtual crack extension evaluation does, however rely on a significantly more complicated setup of the model, so for cases when the J-integral method is applicable, that should be the first choice.

USING THE VIRTUAL CRACK EXTENSION METHOD

The first step is to add a Virtual Crack Extension Method subnode under the **Crack** node used to model the crack. You can only have one **Virtual Crack Extension** subnode under each **Crack** node.

In most cases, the default settings in **Virtual Crack Extension** are sufficient. You may want to adjust the size of the sensitivity region. Its significance is discussed in Virtual Crack Extension Theory. If the crack geometry is such that there may be more than one crack front, you also need make a selection in the **Crack Front** section.

Then, click the **Automated Model Setup** icon ( ) on the **Sensitivity** section toolbar and select the **Create Deformed Geometry and Study** ( ). This will initiate the following actions:

- A **Deformed Geometry** node with subnodes is added in the component. See Deformed Geometry Features for more information.
- A **Control Variable Field** node is added under **Definitions**, having the crack front as selection.
- A new study named **Virtual Crack Extension Study** is added. It contains a Sensitivity study step with an appropriate objective function. In this study all physics interfaces and multiphysics couplings except the current physics interface are disabled by default. For multiphysics fracture mechanics problems, it is possible to enable the required physics interfaces and multiphysics couplings in the **Virtual Crack Extension Study**.

Once the study has been run, the predefined evaluation group **Fracture Mechanics Results** will contain the values of the energy release rate and stress intensity factors.

Stress Intensity Factors

Stress intensity factors are computed from J-Integral and Virtual Crack Extension. For a linear elastic material, there is a relation between the J-integral (or energy release rate) and the stress intensity factors. This relation is, however, only unique when the stress intensity factors are known, and not when the value the J-integral is known:

$$J = G = \frac{1}{E_{\text{eff}}}\left(K_{\text{I}}^2 + K_{\text{II}}^2 + \frac{1}{1-\nu}K_{\text{III}}^2\right) \tag{2-30}$$

The split between K_{I}, K_{II}, and K_{III} is estimated using the displacement field on the crack faces at some distance behind the crack tip.

In Equation 2-30, E_{eff} is an effective Young's modulus, defined as

$$E_{\text{eff}} = \begin{cases} \dfrac{E}{1-\nu^2} & \text{if plane strain} \\ E & \text{if plane stress} \end{cases}$$

This expression is actually only defined for 2D stress states. In 3D, the expression for plane strain is used, since the stress state at the crack tip in a thick solid essentially resembles plane strain.

FREE BOUNDARY IN 3D

At a free boundary, a state of plane stress prevails. Close to the boundary, there will be a transition zone where neither plane stress nor plane strain is applicable. In this region, the stress intensity factors, as computed from the J-integral, may be uncertain. It should be noted, however, that for most metals, the variation in E_{eff} between plane strain and plane stress is about 10%, leading to an error in the computed stress intensity factor in the transition region that is no more than 5%.

UNITS FOR STRESS INTENSITY FACTORS

The SI unit for stress intensity factors is $\text{N/m}^{3/2}$. Other common units are $\text{MPa}\cdot\text{m}^{1/2}$, $\text{psi}\cdot\text{in}^{1/2}$, and $\text{ksi}\cdot\text{in}^{1/2}$. The unit handling system in COMSOL Multiphysics only operates with integer powers of the base units, and can thus not handle stress intensity factors in a consistent manner.

Because of this, the expressions that define the stress intensity factors include a removal of the units, and the stress intensity factors appear as a unitless quantities. The value of a stress intensity factor should be interpreted as having an implicit unit, which is composed from the base units in the current unit system, for example, $\text{N/m}^{3/2}$.

Contact Between Crack Faces

When a crack is defined by an interior boundary, or is located on a symmetry boundary, it is possible to consider contact between the crack faces by adding a **Crack closure** subnode. When such a node is added, all boundaries selected in the **Crack** node are applicable and selected in the **Crack closure** subnode as well, which means that contact is added between all crack faces.

Contact between crack faces is implemented by the penalty method as described in Contact Analysis Theory in the *Structural Mechanics Theory* chapter. By default,

frictionless contact is considered. However, if the crack is defined by interior boundaries, you can also take friction into account.

The kinematics of the contact condition between crack faces differs from that used in general contact analysis and the **Contact** node. It is fully defined by the displacement discontinuity defined by the slit condition of the crack, and thus involves no contact search algorithm. The gap g_n is then given by

$$g_n = \mathbf{n} \cdot (\mathbf{u}_u - \mathbf{u}_d)$$

where $\mathbf{n}$ is the spatial normal to the interior boundary, and $\mathbf{u}_u$ and $\mathbf{u}_d$ are the displacements on the "upside" and "downside", respectively. Similarly, the tangential deformation g_t is given by

$$g_t = -\mathbf{t} \cdot (\mathbf{u}_u - \mathbf{u}_d)$$

where $\mathbf{t}$ is the spatial tangent to the interior boundary. In 3D, there are two components of g_t given by $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively. Friction is based on an incremental formulation, and the incremental slip is given by

$$\Delta g_t = g_t - g_{t,old}$$

where $g_{t,old}$ is the tangential deformation at the previous converged increment. For a symmetric crack $\mathbf{u}_u = 0$ and $\mathbf{u}_d = \mathbf{u}$. Note that no tangential deformation is computed in such cases. Given that the gap and the slip are computed from the displacement jump across an interior boundary, the contact formulation is only valid for small sliding since the same nodal points are always connected even as the bodies deform.

Buckling Analysis

Linearized Buckling Analysis

A linearized buckling analysis can be used for estimating the critical load at which a structure becomes unstable. This is a predefined study type that consists of two study steps: An initial step in which a unit load is applied to the structure, and a second step in which an eigenvalue problem is solved for the critical buckling load.

The idea behind this type of analysis can be described in the following way:

Consider the equation system to be solved for a stationary load $\mathbf{f}$, in case of geometric nonlinearity;

$$K\mathbf{u} = (K_L + K_{NL})\mathbf{u} = \mathbf{f}$$

Here the total stiffness matrix, K, depends in the solution, since the problem is nonlinear. It has been split into a linear part, K_L, and a nonlinear contribution, K_{NL}.

In a first-order approximation, K_{NL} is proportional to the stress in the structure and thus to the external load. If the linear problem is solved first for an arbitrary initial load level $\mathbf{f}_0$,

$$K_L\mathbf{u}_0 = \mathbf{f}_0 \tag{2-31}$$

then the nonlinear problem can be approximated as

$$(K_L + \lambda K_{NL}(\sigma_0))\mathbf{u} = \lambda\mathbf{f}_0$$

where λ is called the *load multiplier*. Here, σ_0 is the stress from the preload, $\sigma_0(\mathbf{u}_0)$.

The nonlinear contribution, K_{NL}, is obtained by assembling only the nonlinear contributions to the first-order approximation of the full stiffness matrix K.

An instability is reached when this system of equations becomes singular, so that the displacements tend to infinity. The value of the load at which this instability occurs can be determined by, in a second study step, solving an eigenvalue problem for the load multiplier λ.

$$(K_L + \lambda K_{NL}(\mathbf{u}_0))\mathbf{u} = 0$$

COMSOL reports a *critical load factor*, which is the value of λ at which the structure becomes unstable. The corresponding deformation is the shape of the structure in its buckled state.

The value of the initial load is immaterial, since a linear problem is solved. If the initial load actually was larger than the buckling load, then the critical value of λ is smaller than 1. It is also possible that the computed value of λ is negative. This signifies that a reversed load will give the critical case.

LIVE AND DEAD LOADS

In some situations, not all loads that act on the structure can vary, and you may want to compute the critical load factor only with respect to the uncertain ('live') load. For example, gravity is often treated as a 'dead' load. The dead loads do, however, affect the stress distribution in the structure, so they cannot be completely ignored. Using the same terminology as above,

$$K\mathbf{u} = (K_{\mathrm{L}} + K_{\mathrm{NL}}^{\mathrm{d}} + K_{\mathrm{NL}}^{\mathrm{l}})\mathbf{u} = \mathbf{f}$$

Here, the superscripts 'd' and 'l', stands for live and dead loads, respectively. By solving two problems, one with the live loads, and one with the dead loads, it is possible to separate their effects:

$$K_{\mathrm{L}}\mathbf{u}_0^{\mathrm{l}} = \mathbf{f}_0^{\mathrm{l}}$$

$$K_{\mathrm{L}}\mathbf{u}_0^{\mathrm{d}} = \mathbf{f}_0^{\mathrm{d}}$$

Now, the following eigenvalue problem is solved to determine the critical load factor:

$$(K_{\mathrm{L}} + K_{\mathrm{NL}}(\mathbf{u}_0^{\mathrm{d}}) + \lambda K_{\mathrm{NL}}(\mathbf{u}_0^{\mathrm{l}}))\mathbf{u} = 0$$

For any load feature, you can specify that is a dead load by selecting the **Treat as dead load** checkbox in the **Linear Buckling** section.

To perform a buckling analysis including both live and dead loads, you need three study steps: two stationary steps, and the linear buckling study step.

1. In the first study step, you solve for the dead loads only. Disable any live loads in the solver.
2. In the second study step, you solve for all loads; the live loads with an arbitrary scaling factor plus the dead loads.

3 Finally, in settings for the **Linear Buckling** study step, you need to point to these two solutions as **Values at linearization** point and **Live loads solution**, respectively.

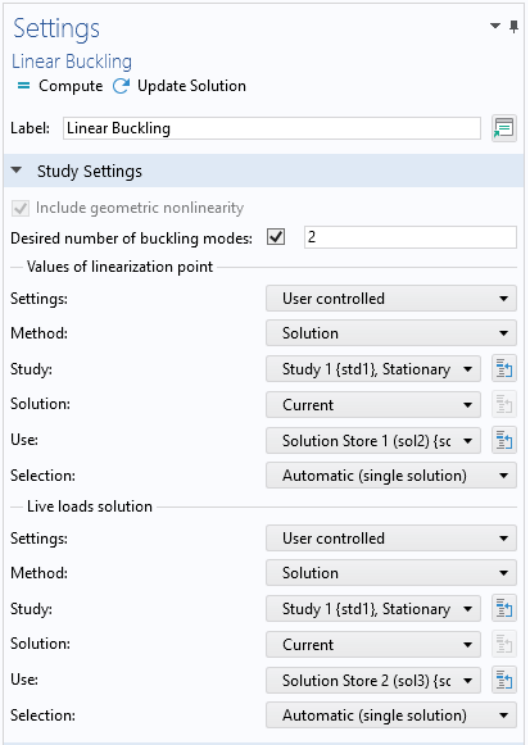


Figure 2-46: Selecting the solutions for a buckling analysis with live and dead loads.

GEOMETRIC NONLINEARITY

Sometimes, the preload case requires a geometrically nonlinear analysis in order to produce the correct state. This means that it is no longer solved using the linear set of equations given by Equation 2-31, but rather

$$(K_{\mathrm{L}} + K_{\mathrm{NL}}(\mathbf{u}_0))\mathbf{u}_0 = \mathbf{f}_0 \tag{2-32}$$

The assumption for the buckling analysis is still that K_{NL} is proportional to the external load, even though this may be disputable for a strongly nonlinear case. K_{NL} is based on the stresses, which must be computed in the same way for both cases, that is, under the same assumption about geometric nonlinearity. The effect is that the stiffness

matrix at the linearization point includes the nonlinear part from Equation 2-32, and the eigenvalue problem is reformulated as

$$((K_{\mathrm{L}} + K_{\mathrm{NL}}(\mathbf{u}_0)) + (\lambda - 1)K_{\mathrm{NL}}(\mathbf{u}_0))\mathbf{u} = 0$$

FOLLOWER LOADS

Loads that depend on the deformation are called *follower loads*. An example of this is a pressure load, since the orientation of the load will depend on surface deformation. Such loads contribute to the stiffness matrix, and can thus affect the buckling load. As a default, all loads in the structural mechanics interfaces are multiplied by the load factor λ in a linear buckling study step.

SYMMETRY AND PERIODICITY

For structures that exhibit symmetries in geometry as well as loads, the buckling modes may still be nonsymmetric. You must take this into account. The easiest method is, of course, just to model the full structure. However, you can reduce the model size and make use of the symmetry properties in using the following methods:

- For a structure with a single symmetry plane, model half the structure, and perform two buckling analyses with different sets of boundary conditions: Symmetry and antisymmetry.
- For a structure with two orthogonal symmetry planes, model one quarter of the structure. Then, perform the buckling analysis with four sets of boundary conditions covering all permutations of symmetry and antisymmetry.
- For a structure that exhibits circumferential periodicity, model one sector, and use a **Periodic Condition** with **Type of periodicity** set to **Cyclic symmetry**.
- For a fully rotationally symmetric structure, you can model an arbitrary sector, and then use a **Periodic Condition** with **Type of periodicity** set to **Cyclic symmetry**.
- For a periodic structure, analyze a unit cell using a **Periodic Condition** with **Type of periodicity** set to **Floquet periodicity**.

For the last three cases, you must repeat the study using a suitable number of azimuthal mode numbers or k-vectors, respectively. There is no way to a priori know which case that has the lowest critical load factor.

Solving for several azimuthal mode numbers or k-vectors can easiest be implemented as an auxiliary parametric sweep using the **Study Extensions** section in the **Linear Buckling** study step node.

When using cyclic symmetry, you may get computed modes having a mode number other than the one you asked for. Such modes are also correct solutions. This phenomenon is an effect of the fact that the trigonometric boundary conditions are only unique up to a factor 2π. The general rule is that if you search for modes with mode number m, you will get all modes with mode number $|m + kN|$. Here, N is the number of sectors for a full revolution, and k is any integer, positive or negative.

Unless you actually want to study modes with a certain mode number, it is still sufficient to solve for one buckling mode only. If the computed mode has an unexpected mode number, that only means that the critical load factor for that mode is lower than for any mode with the mode number you asked for.

If your model is fully rotationally symmetric, then it is a good idea to use a very small sector in the analysis. Not only will you have a low number of degrees of freedom, but since N is then large, you will usually get the modes you asked for. Modes with high mode numbers have complicated patterns that will cause high critical load factors.

If the computation of the predeformation requires a geometrically nonlinear analysis, then geometric nonlinearity must be used also in the Linear Buckling study step.

In this case, it must be assumed that the critical load factor λ is significantly larger than 1.

- Be aware that for some structures, the true buckling load can be significantly smaller than what is computed using a linearized analysis. This phenomenon is sometimes called *imperfection sensitivity*. Small deviations from the theoretical geometrical shape can then have a large impact on the actual buckling load. This is especially important for curved shells.
- It must always be checked that allowable stresses are not exceeded before the buckling load is reached.

- Studies and Solvers and Linear Buckling in the *COMSOL Multiphysics Reference Manual*
- Linear Buckling in the theory section of the *Structural Mechanics Module User's Guide*

- *Bracket — Linear Buckling Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_linear_buckling**
- *Buckling Analysis of a Truss Tower*: Application Library path **Structural_Mechanics_Module/Buckling_and_Wrinkling/truss_tower_buckling**

Nonlinear Buckling Analysis

A linear buckling analysis gives an approximation to the critical load of a structure. To gain more confidence in the results, a full nonlinear analysis where the load is slowly increased can be performed. Such an analysis will, in general, not give a distinct value of the buckling load. Rather, there will be an increasing level of deviation from the linear response. The allowable load is given by some criterion of allowable deformation or stress. In some cases, the nonlinear buckling analysis will stop because a singularity is encountered. This can then be considered as the buckling load.

In a nonlinear buckling analysis, you use an auxiliary sweep, where a parameter for the load is increased. This type of analysis requires significantly more computing time than a linearized buckling analysis.

Modeling Imperfection in a Buckling Analysis

Real structures always contain some level of imperfections in the geometry. Beams are not perfectly straight, plates are not perfectly flat, and so on. For some structures, the real-life failure load is much smaller than the ideal buckling load that would be the result of a linearized buckling study.

There are different ways in which you can take imperfections into account. If the actual imperfections are known, they may even be part of the geometry. More common,

however, is that the analysis is based on the ideal geometry. In this case, using the deformed geometry concept provides a convenient tool for introducing the imperfection.

For more information about deformed geometries, see Deformed Geometry and Moving Mesh and Deformed Geometry Features in the *COMSOL Multiphysics Reference Manual*

You cannot perform this type of analysis with the Layered Shell interface.

If the imperfection is known from some external source, say a standard that stating that a certain curvature must be assumed for a beam, then you can directly enter expressions for such deviations. Do as follows:

1. On the **Definitions** tab, click **Deformed Geometry**, and select **Prescribed Deformation**. Alternatively, you can right-click on the **Component** node and select **Deformed Geometry** > **Prescribed Deformation**.
2. In the added **Prescribed Deformation** node, select the appropriate **Geometric entity level**, and then select the part of the geometry to perturb.
3. Enter expressions for the **Prescribed deformation** in terms of the geometry frame coordinates, for example, `Xg`, `Yg`, and `Zg`.

Another common approach for generating an initial imperfection is to first perform a linearized buckling analysis, and then add one or more suitably scaled buckling modes as initial imperfection. The rationale behind this is that it is reasonable to assume that the structure is sensitive to an imperfection that resembles a buckling mode.

The procedure for entering buckling models as imperfections is also based on a deformed geometry, but is more complicated. For this reason, a special tool is available for setting up such an analysis, as described below.

1. Run a linear buckling analysis, in which you solve for one or more buckling modes.
2. On the **Definitions** tab, click **Physics Utilities**, and select **Buckling Imperfection**. Alternatively, you can right-click on the **Definitions** node and select **Physics Utilities** > **Buckling Imperfection**.

3 In the added Buckling Imperfection node, do the following

a From the **Linear buckling study** list, choose the study from which the imperfection mode shapes are to be selected. Only studies containing a **Linear Buckling** study step are shown in the list.

b Under **Mode selection**, add the buckling modes to include in the **Mode** column. Also, specify a scale factor for each mode in the corresponding field under **Scale factor**.

c To add a **Deformed Geometry** node with necessary subnodes in which the selected sum of buckling modes are used as predeformation, click the **Create** button ( ) in the **Deformed Geometry** section header.

d From the **Study** list, choose the study that is the nonlinear buckling study. The default, and most common case, is **New**, in which case the study does not already exist. You can also select any existing **Stationary** study.

e From the **Load parameter** list, choose the parameter to use as a load parameter for ramping up the load. The parameter must be defined under a **Parameters** node, so you may have to move there to create it. Its purpose is to act as a multiplier to the same load that was used in the linear buckling study.

f Click the **Create** button ( ) in the **Nonlinear Buckling Study** section header to set up the nonlinear study. If **Study** is set to **New**, a new study is created. If an existing study is selected, its settings will be modified. In either case, geometrical nonlinearity will be activated in the study, and a continuation solver will be set up

using the **Load parameter** as auxiliary sweep parameter. The range of the sweep is based on the lowest buckling mode selected in the **Mode selection** table.

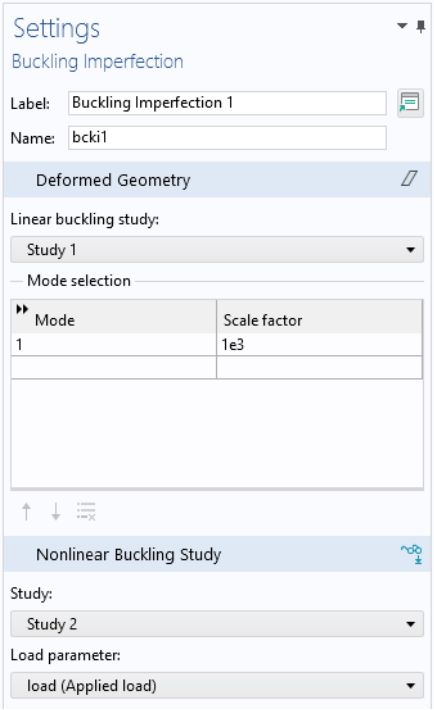


Figure 2-47: Settings in the Buckling Imperfection node.

4 Go back to the physics, and make sure that the loads are multiplied by the load multiplier parameter.

5 Run the nonlinear buckling study.

- When the nonlinear study is generated, the default expression for the auxiliary sweep parameter values is set to

  `<CritFactor>*log(range(1,1,20))/log(15)`

  Here, `<CritFactor>` is the critical load factor in the linearized buckling analysis. This means that the load is increased in 20 decreasing steps, from 0 to about 1.1 times the critical load factor.
- Under Definitions, a **Deformed Geometry** node with one or more **Prescribed Deformation** subnodes is created. These subnodes contain references to special variables containing the selected superposition of buckling modes (Figure 2-48).

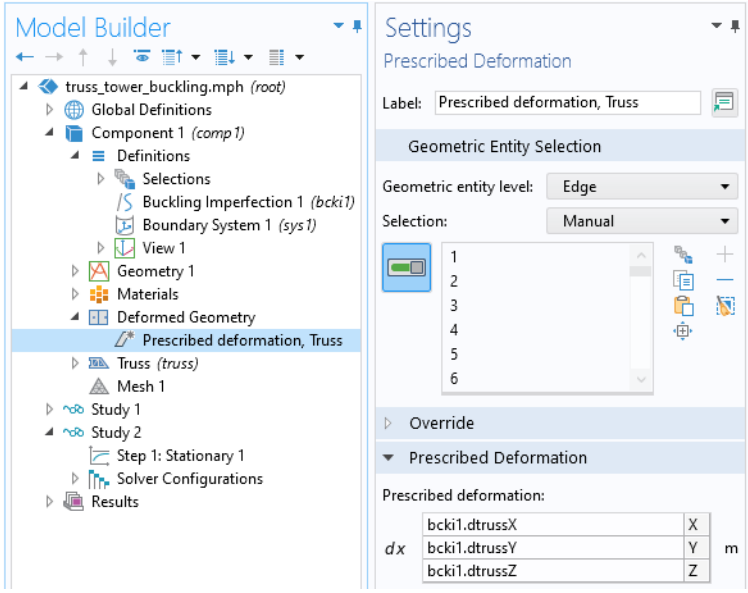


Figure 2-48: The automatically generated deformed geometry settings.

Buckling Analysis of a Truss Tower: Application Library path **Structural_Mechanics_Module/Buckling_and_Wrinkling/truss_tower_buckling**

Performing an Explicit Dynamics Analysis

In this section:

- Introduction to Explicit Dynamics
- Geometry and Meshing for Explicit Dynamics
- Setting up the Physics
- Setting Up and Solving an Explicit Dynamics Study

Introduction to Explicit Dynamics

The time stepping for a structural dynamics analysis, that is a Time-Domain Analysis that includes inertial forces, can be made using either implicit or explicit methods. The fundamental difference between the two is that implicit methods require global equilibrium of forces to be established before advancing the solution in time, while explicit methods do not. Not requiring equilibrium greatly simplifies the computations of each time step for explicit methods, however, it comes with a distinct caveat, the stability of the method is conditional on the size of the time step. The stable time step is determined by factors such as the mesh and material properties. As a consequence, solving a time history using an explicit method typically amounts to taking many cheap but small increments in time, which can be costly for long duration events. This limitation is not shared by implicit methods where, in theory, the size of the time step is only limited by the required accuracy in time. In practice, other factors, such as the convergence of the nonlinear solver used to establish the global equilibrium, limit the step size also for implicit methods.

For many structural problems, implicit methods are the most efficient, robust, and widely used approach, so also in COMSOL Multiphysics. However, for certain conditions implicit methods struggle, for example:

- Severe nonlinearities
- Rapid changes in loads and boundary conditions
- Poorly conditioned system matrices (mass and stiffness).

The above conditions typically results in more nonlinear iterations and smaller time steps, increasing the overall cost of implicit solution. Often, it may also be difficult to

establish equilibrium at all causing a premature termination of the solution. Here is where explicit methods has a clear advantage and can be the preferred choice.

PROBLEMS SUITABLE FOR EXPLICIT DYNAMICS

Problems suitable to solve using explicit time stepping generally involves short duration events, but explicit dynamics can also be used to solve nonlinear quasistatic problems. A summary of typically situations that may warrant the use of explicit dynamics is given below

- Short duration transient loading conditions typically measured in the order of milliseconds. This can range from structural wave propagation at large scales to impact events.
- Complex contact problems with many and frequently changing contact interactions.
- Extreme material nonlinearities caused by, for example, plasticity or damage.
- Geometrical nonlinearity that leads to bifurcation of the solution such as a snap-through.

Many models will include a combination of the above. For example, a postbuckling analysis by definition involves geometric nonlinearity, but the large deformations often also leads to severe plastic deformations, and contact is typically required to handle self-intersections.

- *Bracket — Explicit Dynamics*
- *Objects Falling in a Box*

Geometry and Meshing for Explicit Dynamics

Preparing the geometry and mesh for an efficient and successful explicit dynamics simulation often requires more effort compared to when using implicit solution methods. To a large part this comes from the conditional stability of the time stepping method. Also, the fact that it is preferred to use linear shape functions for the displacements adds requirements on the mesh to obtain an accurate solution.

The **Solid Mechanics, Explicit Dynamics** and **Truss, Explicit Dynamics** interfaces only support linear shape functions.

GEOMETRY

The requirements on the geometry is a consequence of the subsequent requirements on the mesh to control the stable time step. In order to facilitate a good quality mesh for explicit dynamics consider the following:

- Keep the geometry as simple as possible
- Avoid unnecessary details or feature of the geometry not important for the outcome of the simulation
- Make use the **Geometry Cleanup** tool, **Virtual Operations** and other functionality available in the geometry sequence to simplify the geometry.

MESHING

The accuracy and cost of an explicit dynamics simulation is ultimately determined by the quality of the mesh. For example a few small or poor quality mesh elements will dictate the time step taken by the solver; thereby significantly increasing the solution time with little to no gain in accuracy.

Adhering to the following guidelines is good practice that will increase the quality of the simulation results:

- Use an as uniform mesh size as possible
- Avoid small mesh element
  - Small mesh element can significantly reduce the time step.
  - If not possible in limited parts of the model, use Mass scaling.
- Avoid poor quality mesh elements
  - Poor aspect ratio and large skewness have negative effects and reduces both the stable time step and the accuracy of the results.
- Use high fidelity mesh elements where possible
  - Hexahedron is the preferred mesh element type in 3D
  - Quadrilateral is the preferred mesh element type in 2D
  - Avoid simplex mesh elements, tetrahedrons and triangles, if possible.

The above recommendations on mesh element types follow from the use of linear shape functions for the displacements. It is well-known that linear simplex elements have poor accuracy and are prone locking, and therefore should be used sparingly. Also, hexahedra and quadrilaterals exhibit both shear and volumetric locking. However, their accuracy is much improved by using reduced integration. While better then simplex elements, the use prism elements in 3D should also be minimized. Even

with reduced integration they exhibit locking and can thus lead to an overly stiff model response.

See Using Reduced Integration for more information.

Setting up the Physics

Time-explicit dynamic analysis is supported by specialized physics interfaces with settings and functionality tailored for the explicit solution method:

- **Solid Mechanics, Explicit Dynamics**
- **Truss, Explicit Dynamics**

The main differences from the Solid Mechanics and Truss interfaces are:

- Provides the explicit solver with estimates of the time step
- Includes the **Artificial Viscosity** and **Mass Scaling** features
- Only linear shape functions for the displacement field
- Reduced integration used as default
- Weak constraints are not available
- Some physics features are not available

See documentation of the

- The Solid Mechanics, Explicit Dynamics Interface
- The Truss, Explicit Dynamics Interface

MATERIAL MODELS

The explicit dynamics interfaces only support the isotropic materials models included in **Linear Elastic Material** and **Hyperelastic Material**. Most additional material behavior, including nonlinear effects, included by adding subnodes such as **Damping**, **External Strain**, **External Stress**, **Plasticity** and **Damage** are available and can be used without restriction.

General recommendations to consider when adding material models:

- The time step estimate is based only on the elastic material behavior. Be careful when adding subnodes that may cause a stiffening effect.
- Any source of damping affects the stable time step, see Damping.
- Do not add complex material behavior to domains where they are of little importance. For example, **Plasticity** can add a significant cost when solving the model also if there is no plastic deformations since the yield condition still needs to be checked every times step.

Anisotropic behavior can be included by adding one or several **Fiber** features to the material model.

LOADS AND CONSTRAINTS

General recommendations to consider when adding loads and boundary conditions:

- Point loads and constraints tend to excite hourglass deformations if reduced integration is used. Avoid if possible.
- Be careful when adding boundary conditions that add stiffness to the model like **Spring Foundation**. The added stiffness is not accounted for when in the stable time step estimate and can thus cause the solution become unstable.
  - Other boundary conditions that add stiffness include the limited displacement option of **Prescribed Displacement** and **Interior Contact**.
  - Pair conditions like **Bolt Threaded Contact**, **Contact** and **General Contact** also add stiffness.
- Volume forces can be costly to evaluate. For example, **Gravity** often have negligible effects on short duration events and can often be omitted to reduce the solution time.
- Apply transient loads smoothly to avoid shocks were not expected.

Constraints added by, for example, **Continuity**, **Prescribed Displacement**, **Roller**, or **Symmetry** are always implemented as pointwise constraints for explicit dynamics. The constraint handling for explicit solvers in COMSOL Multiphysics has limitations compared to the implicit solver. Most notably, a constraint must not depend on the solution; that is, nonlinear constraints are not supported. A constraint can, however,

depend on time. These restrictions are seldom any real limitation for structural mechanics, but two cases that can occur when geometric nonlinearity is enabled are:

- Identity pairs selected by **Continuity** must use the material frame for both source and destination

A constraint can be depend on the material coordinates but not the spatial coordinates..

See also Explicit Dynamic Contact Analysis.

INITIAL CONDITIONS

The time step size restriction of an explicit solver means that it is costly to solve long duration events. If the majority of the event is of little interest it can be excluded from the explicit dynamics simulation by instead starting close the time of interest with initial conditions to account for the preceding time.

A common case where this applies is for a drop test from a specified height under gravity accelerations. Rather than solving for the entire free fall, place the object close to ground and apply an initial velocity corresponding to the free fall. For more complicated cases, it is possible to seamlessly use any solution from another study step as initial conditions to the **Explicit Dynamics** step.

The **Explicit Dynamics** interfaces can be used with most standard study steps available in COMSOL Multiphysics. Common study steps to define initial conditions for the explicit dynamics include:

- The **Stationary** study step to compute static initial values. For example, initiate contact and friction.
- The **Bolt Pretension** study step to initialize bolted connections.
- The **Time Dependent** study step to use implicit time-stepping initially.

A **Time Dependent** study step can also be added after the explicit dynamics step to switch back to implicit time-stepping.

STRUCTURAL COUPLINGS

The **Solid Mechanics, Explicit Dynamics** and **Truss, Explicit dynamics** can be coupled using the **Embedded Reinforcement** multiphysics coupling.

See Modeling Embedded Structures and Reinforcements for more details.

MULTIPHYSICS

The **Explicit Dynamics** interfaces and study step have limited support for multiphysics simulations. Supported multiphysics i capabilities include

- The **Adiabatic Heating** feature to model a local temperature effects caused by dissipation.
- The **Phase-field Damage** coupling to model the evolution of damage and cracks. Here the algebraic phase field equations added by the **Phase Field in Solids** interface are solved implicitly while time-stepping.
- Values of any dependent variable like temperature not solver for by the explicit dynamics study step can be taken from another solution.

EQUATION-BASED MODELING

The **Explicit Dynamics** interfaces support equation-based modeling tools. However, note that no equations or dependent variables added are accounted for in the automatic step size estimate for explicit solvers.

Setting Up and Solving an Explicit Dynamics Study

To use explicit time-stepping for a model that includes any physics interface for explicit dynamics, add an **Explicit Dynamics** study step to a new or existing study.

Using any other study step with the explicit dynamics interfaces will generate a default solver sequence with an implicit method.

The **Explicit Dynamics** study step creates a solver sequence with the Time-Dependent Solver configured to use an explicit solver type for time stepping. By default, the **Verlet** method is used, but other methods are also available. The Verlet method is similar to as so-called central difference scheme, often used for explicit dynamics. Two important properties of this method is that it updates velocities at the half-step of each increment,

and that it only requires one evaluation of the internal forces per step. The latter is a key to the efficiency of the method when compared to alternative methods.

Also important for an efficient solution is the use of a lumped, or diagonal, mass matrix. When a lumped mass matrix is used, accelerations are trivial to compute with no need to invert a large mass matrix, since the system equations are effectively decoupled. A lumped mass matrix is used by default when adding the **Explicit Dynamics** study step. An alternative is to instead use a consistent mass matrix, that is, directly use the mass matrix assembled form the weak equations. In this case the solver will assemble the mass matrix and invert it by a linear solver every time step in order to compute accelerations.

See documentation of the Explicit Dynamics study step for more information about available settings.

THE STABLE TIME STEP

A fundamental property of any explicit time-stepping method is its conditional stability; the allowable time-step size is limited. A too large time step will cause significant numerical errors to develop and the solution will grow unbounded to an eventually failure. The time step by which the solution remains bounded is referred to as the *stable time step*, or the *critical time step*.

The stability of any explicit method is given by the highest natural frequency, ω_{max}, of the discretized model equations solved. The stable time step is then

$$\Delta t_{stb} = \beta \frac{1}{\omega_{max}}$$

where the factor β depends to the time-stepping method. For the Verlet method, $\beta = 2$.

To find the true and optimal value for Δt_{stb}, would require an eigenfrequency analysis of the entire model to find ω_{max}. Given that the number of modes is equal to the number of degrees of freedom of the model, the cost of such an operation is large. For nonlinear models, ω_{max} also changes with time and deformation of the structure, making such an approach for finding a stable time step unfeasible.

A better strategy is to approximate ω_{max} as the maximum of the highest natural frequency, $\omega_{max,e}$, of all mesh elements in the model. Such approximation will always overestimate ω_{max}, that is

$$\omega_{\max} < \max\ \omega_{\max,\,e}$$

or

$$\Delta t_{\text{stb}} < \beta \min \frac{1}{\omega_{\max,\,e}} = \beta \min t_{\text{cell},e}$$

where $t_{\text{cell},e}$ is called the cell time. It can be interpreted as the shortest time it takes for a wave to pass for a computational cell, that is, a mesh element. We can thus define

$$t_{\text{cell},e} = \frac{1}{2} \frac{h_e}{c_{\max,\,e}} \tag{2-33}$$

where h_e is the characteristic size of the mesh element and $c_{\max,e}$ is the maximum wave speed of the material.

The actual value of $t_{\text{cell},e}$ depends also on additional factors such as integration order, lumped or consistent mass matrix, mesh element type, and mesh element quality.

All material models in the explicit dynamics interfaces provide a value for $c_{\max,e}$ computed from the material properties. If geometric nonlinearity is included, $c_{\max,e}$ is defined from the current (deformed) wave speed with respect to the material frame

The wave speed $c_{\max}$ does not account for the effects of inelastic deformations such as plasticity or damage.

The explicit dynamics interfaces provide the time-dependent solver with expressions to evaluate $t_{\text{cell},e}$ over all domain mesh elements while time-stepping to update the time step Δt used. The actual time step used is defined as

$$\Delta t = f_{\text{time}} \beta \min t_{\text{cell},e}$$

where f_{time} is global time step safety factor used scale the time step. It can be used to, for example, reduce the time step in case there are stiffness and mass contributions added to the model not accounted for in the definition of $t_{\text{cell},e}$.

Damping

The fact that the velocity for the Verlet method lags by one half-step when evaluating the internal forces leads to damping having a negative impact on its stability. It can be

shown that the stable time step is reduced by a factor proportional to the critical damping ratio ξ. When damping is added to a physics interface, the cell time estimate in Equation 2-33 is therefore modified to

$$t_{\text{cell},e} = \frac{1}{2}\frac{h_e}{c_{\max,e}}\left(\sqrt{\xi_e^2 + 1} - \xi_e\right)$$

where ξ_e is the critical damping ratio of the mesh element.

Since the highest natural frequency of a mesh element, $\omega_{\max,e}$, determines $t_{\text{cell},e}$, the critical damping ratio ξ_e is also defined with respect to $\omega_{\max,e}$. Consequently, stiffness proportional damping will have a particularly unfavorable impact on the time step.

Common stiffness proportional damping contributions include:

- The stiffness part of Rayleigh damping
- Viscous damping

Mass scaling

It can be observed that the time step estimate for each mesh element provided by Equation 2-33 decreases with the size of the mesh elements. Hence it is likely that the smallest mesh element will determine the time step used by the solver. For this reason, small mesh elements should be avoided, as emphasized in Geometry and Meshing for Explicit Dynamics. However, for realistic geometries it may not be possible to avoid small elements completely. Considering Equation 2-33, an alternative to increase the times step is then instead to decrease the wave speed $c_{\max,e}$.

Consider a 1D bar, where the wave speed is

$$c = \sqrt{\frac{E}{\rho}}$$

To decrease c, we can either decrease the stiffness $\boldsymbol{E}$ or increase the density ρ. Changing the stiffness of the material is likely to have a significant impact structural and material behavior and is therefore seldom recommended. Adding more mass by increasing ρ is likely to have less impact, and is therefore a viable option for limiting the negative impact of small mesh elements on the solution time. This technique is referred to as *mass scaling*.

To include mass scaling in an explicit dynamic analysis, add a **Mass scaling** node and select the domains where to add mass. The amount of added mass is specified by entering a target value for the cell time at zero deformation, $t_{\text{cell},0}$. An artificial density ρ_a is then defined by scaling the physical density ρ_0 assigned to the material,

$$\rho_a = \left[\max\left(\frac{t_{\text{target}}}{t_{\text{cell},0}}, 1\right)^2 - 1\right]\rho_0$$

where t_{target} is the target value entered. The mass added by **Mass scaling** only contributes with inertial forces and will not affect volume forces due to frame accelerations added by, for example, gravity.

The artificial density ρ_a is constant in time also when geometric nonlinearity is included. The time step used by the solver can therefore be smaller than βt_{target} when there are large deformations.

- A variable `<phys>.cellTime0` is provided to identify mesh element with the smallest cell time at zero deformation. There is also a result template for each physics interface for this purpose.
- The variable for the artificial density is called `<phys>.rhoa`
- The kinetic energy added by mass scaling can be inspected variables `<phys>.Wka` and `<phys>.Wka_tot`.

Mass scaling is a powerful tool for explicit dynamics to reduce the computational cost of a simulation. It is especially attractive when using explicit dynamics for quasistatic analysis of slow process like metal forming where inertial effects are of less importance. When used for true a true dynamic analysis, be extra careful and make sure not to add artificial mass to components critical to the dynamic response. Regardless of use case, it is good practice to always verify the effects of mass scaling prior to running the solution by

- Inspecting the distribution of the artificial mass.
- Comparing the total amount of artificial mass added to the physical mass of the model.

After running the solution, compare the artificial kinetic energy to other energies in the model.

OUTPUT AND RESULTS WHILE SOLVING

Storing output too frequently will damage the computational efficiency of the time-explicit method. To manage this, be careful when selecting output times in **Explicit Dynamics** study step, and plan in advance when output is of interest.

There are different alternatives for how to store output times in the Time-Dependent Solver node. If the **Steps taken by solver** option is selected, also consider at what sampling rate to store the output.

Explicit dynamics is suitable for solving large-model structural mechanics problems since it requires less memory when compared to implicit methods. Storing the solution for large models with a desired time resolution can result in huge data files. To circumvent this, an alternative is to use the Store in Output option of the **Explicit Dynamics** study step. Using the **Selection** option in **Output** allows you to store data only for predefined selections. The list of sections can, for example, include a subset of domains of interest, boundaries of a symmetry plane, and a number points. When postprocessing the data, the best results are obtained by adding the same selections to the dataset used.

For true dynamic problems, the output sampling rate may be dictated by the required accuracy to resolve a signal in time to avoid aliasing. Recall the Nyquist theorem that specifies the minimum data sampling rate required to resolve a signal is two times the frequency of interest. If this leads to too may output times, using **Probes** for selected points of interest stored at times taken by the solver is an alternative. However, using too many probes can be expensive, especially when adding domain or boundary probes.

When using **Results While Solving** with explicit time-stepping, updating plot groups is often more costly then advancing the solution one time step. Therefore, only update plots at **Time stored in output** unless required for debugging the behavior of a model.

Performing a Random Vibration Analysis

INTRODUCTION

Sometimes, the loads on a structure are random in nature. An example is the wind load on a tower. In addition to the average wind, there are gusts caused by the turbulence in the flow. The gusts will give a time-dependent excitation, which can induce a dynamically amplified response in the structure. The frequency and amplitude of these gusts will be randomly distributed. If the structure is large, the peaks in the wind speed may not even occur synchronously at locations far from each other.

If the pressure is measured, it is possible to give a statistical representation of the wind load. The statistical representation of an input like a force to be used in a random vibration analysis is the power spectral density (PSD), which is a function of the frequency.

Similarly, the computed results quantities are represented by their PSD, from which it is possible to derive a root-mean-square (RMS) value.

There are two main cases when a random vibration analysis is useful. In addition to the scenario outlined above, it is common that design standards (in particular for electronic components and devices) include requirements on random vibration testing. To simulate such a test, a random vibration analysis can be performed. In this case, the prescribed PSD of the excitation is a simplified envelope intended to cover a multitude of loading conditions.

For a detailed theory, see Random Vibration Theory.

Correlation

When several loads act on a structure, they can have different degrees of interdependency. This is described by a statistical measure of their correlation. There are two important special cases of correlation, which significantly simplify the analysis.

If the loads are random in nature and completely independent of each other, they can be described as being ***uncorrelated***. This would be the case when, for example, two separate drilling machines are used simultaneously at some distance from each other.

If several loads have the same source, they can be *fully correlated*. Vibration testing is a common example. In such case, several points on the structure are attached to the same shaker table. Thus, they will experience identical (but random) acceleration histories. Another example could be when a certain load is decomposed into components along different coordinate axes. Clearly, the X- and Y-components are just scaled versions of the total load, and thus they are fully correlated.

In a general case, the correlation between two loads is a function of both the distance between their positions and the frequency. The input to the analysis includes not only the PSD of each load but also their cross-correlation.

Spatial Variation of Random Loads

The PSDs and cross-correlation matrices are assumed to be functions of frequency only. It is not allowed to write expressions that are functions of the spatial coordinates. The expression for the load inside the physics interface, however, can have such dependencies as long as they appear as a pure multiplier to the spectrum.

In principle, however, the PSD and even the full cross-correlation functions could be considered to vary continuously over a structure, and thus be functions of the spatial coordinates. An example of this could be the wind pressure on a high tower.

If you want to model such situation, you need to split the loaded region into smaller parts, each with its own constant spectrum definition. Note that the number of off-diagonal cross-correlation functions increases quadratically with the subdivision.

SETTING UP A RANDOM VIBRATION ANALYSIS

Random vibration analysis is based on a modal representation of the structure. It represents a type of mode superposition and relies on the reduced-order model (ROM) concept.

In principle, you can perform random vibration analysis on any reduced-order model that is of the Frequency Domain, Modal type. This requires a number of settings and studies run in an appropriate order.

It is significantly more convenient to start by adding a **Random Vibration (PSD)** study, either from the **Add Study** window or from the **Select Study** page in the Model Wizard.

The **Random Vibration (PSD)** study is not a study in itself; rather, it adds a number of nodes to the Model Builder tree to facilitate a random vibration analysis. These nodes serve the purpose of setting up the ROM and providing input data to the random vibration analysis.

Two studies are added:

- One study with an **Eigenfrequency** study step for computing the eigenfrequencies and corresponding eigenmodes.
- One study with a **Model Reduction** study step, in which the ROM is created. It references the eigenfrequency study, as well as an embedded **Frequency Domain** study step. The latter can mainly be considered as a placeholder which is a mandatory input to the model reduction. In case you have already computed the eigenmodes, you can change the setting of the **Training study** in this node to point to the corresponding existing eigenfrequency study, and then delete the newly generated one.

The **Model Reduction** study step is set up to be suitable for a random vibration analysis. In particular, it is required the **Ensure reconstruction capability** is selected.

Under **Global Definitions**, a **Reduced-Order Modeling** node becomes available. It contains three subnodes:

- A **Global Reduced Model Inputs** node, in which you define the control parameters for the ROM. All loads that represents random excitations must have a value multiplied by one of such parameters.
- A **Frequency Domain, Modal Reduced-Order Model** node. This is a placeholder for the ROM to be created.
- A **Random Vibration** node. Here, you prescribe the PSD functions and the cross-correlation functions, if needed. They can be function of frequency only (the internal variable `freq`); it is not possible to prescribe a coordinate dependency.

WORKFLOW

Here, you find recommended steps to perform a random vibration analysis.

1. Add a **Random Vibration (PSD)** study.
2. In the **Global Reduced Model Inputs** node, add all required control parameters. The values that you assign to the parameters are not important. Control parameters are used in the same manner as ordinary parameters in expressions for the loads in the physics interface. You need as many control parameters as you have different PSD functions.

3 In the physics interface, create the loads that should have a random variation. Typically, the value is just the name of a control parameter, even though it is possible to also add a scalar multiplier to it.

- The reduced-order models on which the random vibration analysis is based are not unit aware. Always use the base units from the current unit system in the context of a random vibration analysis.
- Only 'Neumann type' loads like forces and pressures can be given a random excitation. It is not possible to use 'Dirichlet type' loads like prescribed displacements, velocities, or accelerations. This is similar to other mode superposition methods.

  For the common case that a number of support points have the same acceleration PSD, the support acceleration can be replaced by a frame acceleration load, using the **Base Excitation** feature. This is how vibration testing is analyzed. See also Absolute Versus Relative Accelerations.

4 If you want any scalar outputs, define them as **Variables** under **Definitions** in the component. You can use, for example, probes or functions like `at3()` to create scalar outputs. It is also possible to compute scalar results using, for example, **Point Evaluation** during result presentation. Such evaluations will, however, require a larger computational effort than scalar outputs that are part of the ROM definition.

5 In case you want to reuse a previously computed eigenfrequency study, go to the **Model Reduction** study step. Change the settings for **Training study** in the **Model Reduction Settings** section to point to the correct eigenfrequency study.

6 Run the study containing the **Model Reduction** study step. It will automatically create a ROM that can be used for all further evaluations.

7 Define the PSD and cross-correlation functions required for representing the loading as functions under **Global Definitions**. These functions should depend on frequency only.

It is common that an input PSD is provided in terms of straight lines in a log-log diagram of PSD value versus frequency, *f*. To mimic this behavior, you can use an interpolation function (say `int1`) where you enter pairs of log(*f*) and log(PSD) values. You then reference this function through an expression like `exp(int1(log(freq[1/Hz])))`. This will provide a linear interpolation in a log-log space.

8 Go to the **Random Vibration** node. Select the appropriate **Correlation type**.

9 For each control parameter, enter the corresponding spectrum function. The argument to the function should be the built-in frequency variable `freq`.

10 If the correlation type is **Cross-correlated**, enter also the off-diagonal part of the **Correlation matrix**.

11 To make these settings available for result evaluation, run **Update Solution** for the model reduction study.

RESULT EVALUATION

For presentation of random vibration results, you can use the standard features under **Results**, with the provision that the quantity you are studying is a linear function of the displacements. You can use any linear expression, not only built-in variables.

In addition to linear variables, there is also a special case where the RMS of quadratic forms can be computed.

The evaluation of random vibration results is done through special operators, listed in Table 2-19, which are defined by the **Random Vibration** node (with a tag `rvib1` in this example).

TABLE 2-19: RANDOM VIBRATION OPERATORS.

OPERATOR	SYNTAX	EXPLANATION
psd	rvib1.psd(expr)	Evaluates the PSD of an expression for the given frequency
cross	rvib1.cross(expr1, expr2)	Evaluates the cross-correlation between two expressions for a given frequency,

TABLE 2-19: RANDOM VIBRATION OPERATORS.

OPERATOR	SYNTAX	EXPLANATION
rms	rvib1.rms(expr, f1, f2, N)	Evaluates the RMS value of an expression within the frequency interval [f1, f2]. N points are used for integrating over the interval.
rms	rvib1.rms(expr)	Evaluates the RMS value of an expression. The frequency interval, as well as the integration settings are taken from settings in the corresponding Random Vibration node.
m	rvib1.m(K,expr, f1, f2, N)	Evaluates the K*th* order moment of the PSD. Thus, rvib1.rms(expr, f1, f2, N) = sqrt(rvib1.m(0,expr, f1, f2, N)). The maximum supported moment order is K=4.
m	rvib1.m(K,expr)	Evaluates the K*th* order moment of the PSD. The frequency interval, as well as the integration settings are taken from settings in the corresponding Random Vibration node. The maximum supported moment order is K=4.
q2	rvib1.q2(expr, f1, f2, N)	Evaluates the RMS of a quadratic form, for example the square of an effective displacement, or the square of the von Mises equivalent stress. The expression must not contain time derivatives like velocities or accelerations. Rather than rvib1.q2(utt^2,…), you can write rvib1.q2(u^2*(2*pi*freq)^4,…). The quadratic form itself must be real-valued. This means that evaluation using the von Mises stress can only be done if the material properties are real-valued. In particular, it should be noted that loss factor damping implies a complex-valued stiffness.
q2	rvib1.q2(expr)	Evaluates the RMS value of a quadratic form. The frequency interval, as well as the integration settings are taken from settings in the corresponding Random Vibration node.

TABLE 2-19: RANDOM VIBRATION OPERATORS.

OPERATOR	SYNTAX	EXPLANATION
q2sq	rvib1.q2sq(expr, f1, f2, N)	Evaluates the square root of the RMS of the quadratic form. The norm of the quadratic form is used as input. See q2 operator above for more information. The relation between these two operators is: rvib1.q2sq(sqrt(expr),…) = sqrt(rvib1.q2(expr,…))
q2sq	rvib1.q2sq(expr)	Evaluates the square root of the RMS of the square root of a quadratic form. The frequency interval, as well as the integration settings are taken from settings in the corresponding Random Vibration node.

The operator `q2sq()` is a convenience version of the `q2()` operator. Some important quantities have special variables, intended for use in the `q2sq()` operator, see Table 2-20.

TABLE 2-20: PREDEFINED VARIABLES FOR USE WITH THE Q2SQ OPERATOR.

VARIABLE	DESCRIPTION
<phys>.mises_rv	von Mises equivalent stress
<phys>.disp_rv	Norm of displacement
<phys>.vel_rv	Norm of velocity
<phys>.utt_rv	Norm of acceleration
<phys>.rot_rv	Norm of rotation
<phys>.rotvel_rv	Norm of rotational velocity
<phys>.rotacc_rv	Norm of rotational acceleration

You cannot use ordinary variables like `<phys>.mises` or `<phys.disp>` as arguments to the `q2sq()` operator. The reason is that the argument must never evaluate to zero.

It is convenient to plot the PSD of a result quantity computed, for example, at certain point in the structure for a given frequency range. Such evaluation sweeps over the frequency can be defined in a Global Evaluation Sweep node.

The RMS of a result quantity does not give direct information about its peak value. Formally, probability theory says that for a long enough process, the peak value can be

arbitrarily large. In practice, it is common to assume that the peak value is three (or sometimes four) times larger than the RMS value. There are good reasons for this practice.

- The input signal is usually bounded for physical reasons. It is not a true Gaussian distribution.
- For testing, the shaker table accelerations are limited, and this is even expected in the requirements on the signal.
- The whole random vibration analysis assumes linearity. Nonlinear effects will typically limit the response.

Absolute Versus Relative Accelerations

It is common that the input to a random vibration analysis is a PSD for a base acceleration. Since it is not possible to directly prescribed nonzero displacements (or accelerations) at constrained points, the standard approach is to replace the base acceleration with an inertia load on the structure. This can be seen as a frame transformation, where the analysis is made in a coordinate systems that is connected to the fixture. This change of frame does not affect for example stress or strain results.

The computed accelerations, however, are measured relative to the support points, not relative to a room fixed frame. In most cases, you are interested in the absolute accelerations, that is what would be measured by an accelerometer. When the **Base Excitation** feature is used to describe the acceleration, special variables containing the absolute (room fixed) accelerations are also available.

- *Bracket — Random Vibration Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_random_vibration**
- *Random Vibration Analysis of a Deep Beam*: Application Library path **Structural_Mechanics_Module/Verification_Examples/random_vibration_deep_beam**

Performing a Response Spectrum Analysis

INTRODUCTION

Response spectrum analysis is used to approximately determine the structural response to short nondeterministic events like earthquakes and shocks. The idea is that the event is characterized by the peak response that it would give a single degree of freedom (SDOF) oscillator having a certain natural frequency and damping ratio. The response value is provided as a function of the natural frequency of the oscillator.

If the dynamic structural response is viewed as a mode superposition, each eigenmode within the given frequency range acts as an SDOF oscillator, with a peak amplitude that is known from the spectrum. There is, however, no information about when each mode reaches its peak value. The most conservative assumption is that all modes should be combined using the individual peak values. Using such approach will in most cases result into an extremely conservative design. For this reason, other summation methods have been developed. When performing a response spectrum analysis, you will have access to several such methods.

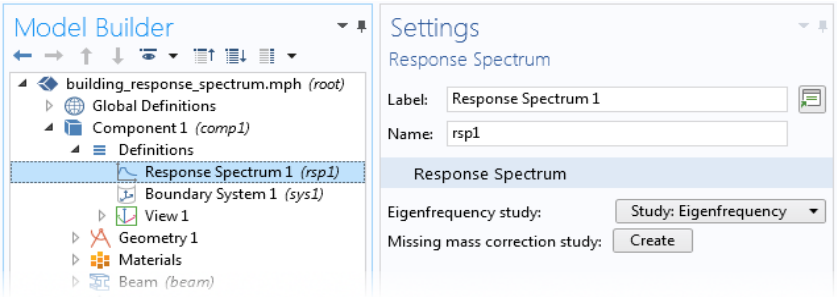
SETTING UP A RESPONSE SPECTRUM ANALYSIS

The response spectrum analysis is not a study type. The computations are performed during result evaluation. You enter the methods and parameters in the settings for the **Response Spectrum 2D** or **Response Spectrum 3D** datasets. These datasets require the following:

- An eigenfrequency solution, where the eigenmodes to be included in the response spectrum analysis have been computed. All computed eigenmodes are used. If you want to filter out a particular set of modes, you can add a **Combine Solutions** study step after the **Eigenfrequency** study step. Such a filter can for example be based on the effective modal mass, so that only modes which contribute significantly to the mass are included.

The time to evaluate a result when using a response spectrum dataset varies almost quadratically with the number of eigenmodes you include.

- A corresponding set of modal participation factors. To generate them, add a **Response Spectrum** node under **Definitions** in the component. If you have computed the eigenfrequency prior to adding the **Response Spectrum** node, you need to perform an **Update Solution**.



- In some response spectrum evaluation methods, you are required to compensate for the mass that is not represented by the included eigenmodes. In order to do so, a number of special stationary load cases must also be computed. You can set up all nodes required in the model builder tree by clicking the **Create missing mass correction study** ( ) button on the **Response Spectrum** section header in the **Response Spectrum** node settings.
- In case you want to sum the response spectrum results with another load case, for example representing self-weight, you also need to solve a separate study for that case. This is an independent step, which can be done at any stage prior to the evaluation.

Adding a Response Spectrum Study

In the **Add Study** dialog, there is a study type called **Response Spectrum**. When you select it, two things happen:

- A study, containing a single **Eigenfrequency** study step is created.
- A **Response Spectrum** node is added under **Definitions** in the first component which contains at least one structural mechanics physics interface. In this node, the **Eigenfrequency study** list will be initialized to point to the **Eigenfrequency** study step that was just created.

If you already have computed the eigenfrequencies of a structure, and then want to perform a response spectrum evaluation, there is no need to add a new study. Just add a **Response Spectrum** node under **Definitions-** > **Variable Utilities**, and then do an **Update Solution** to make the new variables available.

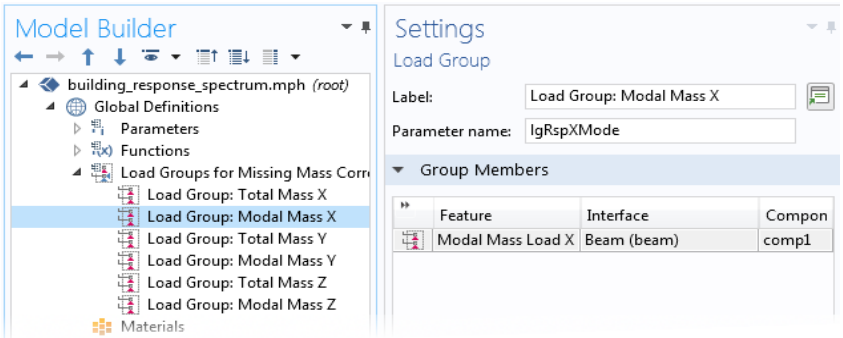
Missing Mass Correction

You create a missing mass correction study from the **Response Spectrum** node by clicking the **Missing mass correction study**: **Create** button.

!	It is not possible to undo this operation. To revert, you will have to manually delete all the nodes which were created.

A number of items are then created in the model tree:

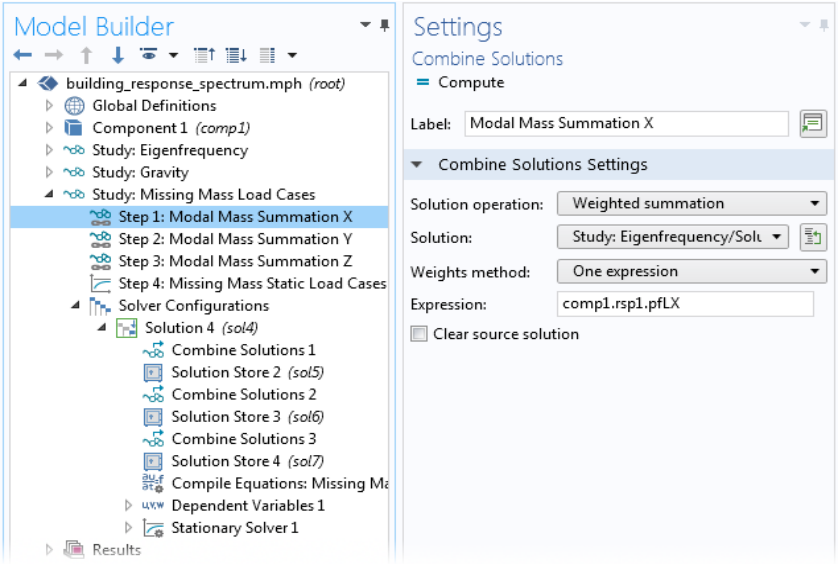
- A set of load group nodes are created under **Global Definitions**. There are two load groups for each spatial direction. The load group nodes are placed under a common group named **Load Groups for Missing Mass Correction**.



  The **Parameter name** of the load group is reserved, and you should not modify it.

- A new study named **Study: Missing Mass Load Cases** is created. This study contains three or four study steps, depending on the space dimension. First, there is one **Combine Solutions** study step for each spatial direction. In these steps, a weighted sum of the eigenmodes is computed. The weights are the modal participation factors in each direction. This gives a measure of the mass that the eigenmodes represent. These study steps must reference the eigenvalue solution, including any subsequent filtering. If you change the study from which the eigenmodes are to be taken, you must also change the choice in the **Solution** list in all of the **Combine Solutions** nodes.

The final study step in the study is a stationary study step, where each load group is solved separately by adding one load case per load group.

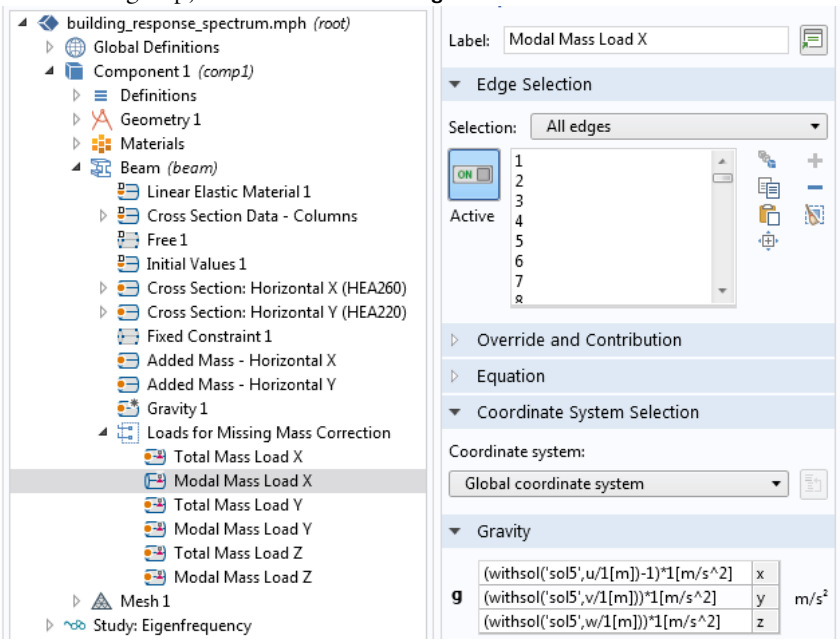


If the physics interface has loads other than those automatically generated for missing mass correction, you need to make sure that those loads are not solved for in the **Missing Mass Static Load Cases** study step. There are several ways of doing that. You can, for example:

- Disable the other load nodes temporarily in the model tree before solving.
- Disable them in this study, by using **Modify model configuration for study step** in the **Physics and Variables** section in the settings for the study step. This is the preferred method.
- Assign the other loads to a new load group.

- In each structural mechanics physics interface in the component, a set of **Gravity** nodes are added. There are two such nodes for each spatial direction. Each gravity node is connected to a corresponding load group. Half of the loads are pure gravity loads, used when only together with the **Static ZPA method**. The other set of loads, which are used in the **Missing mass method**, are referencing the combined solutions

as the depend on the computed eigenmodes. The gravity nodes are placed under a common group, named **Loads for Missing Mass Correction**.



SETTINGS FOR THE RESPONSE SPECTRUM DATASETS

Defining the Spectra

You input the spectra as functions under **Global Definitions**. Usually, either an interpolation function, or an analytic function, or a piecewise function is used.

There are two common methods to describe the design response spectra: either as function of frequency, or as function of period time. Use the **Depends on** setting to control this.

The handling of units in the functions are nonstandard:

- The unit of the argument is ignored. The function is called with either frequency (hertz) or period time (seconds) as defined by the setting in **Depends on**.
- The unit of the function is not checked. The function will, however, be scaled to model units, so if you, for example, enter the unit `mm/s^2` for an acceleration spectrum, the function will (in an SI system) be scaled by 1/1000. You would get the same effect if the entered unit is inconsistent (for example, `mm`).

The spectra you enter have the following correlation to the spatial directions:

- In 2D, you give two spectra: one horizontal and one vertical. The horizontal spectrum acts along the global *X*-axis, and the vertical one acts along the *Y*-axis.
- In 3D, the vertical spectrum always acts along the global *Z*-axis.
- In 3D, you can supply two different horizontal spectra, called primary and secondary horizontal spectrum. Those spectra act in two orthogonal directions, which by default coincide with the global *X* and *Y* directions. By giving a nonzero value to **Primary axis rotation**, you can make the primary spectrum act in an arbitrary direction in the *XY*-plane.
- Some **Spatial combination** methods in 3D (CQC3 and SRSS3) assume that the two horizontal spectra are equal except for an amplitude scale factor. In such case, you only provide one spectrum together with a **Secondary horizontal spectrum scale factor** (value between 0 and 1).

Mode Combination

The combination of the eigenmodes is the core of the response spectrum methods. Most commonly the combination is done in two passes: first the response to the excitation in each spatial direction is determined, and then a total response is computed by combining the spatial responses. However, for certain methods the total response is computed in one pass.

First, you select a **Spatial combination** method. If it is **SRSS** or **Percent method**, you also select a **Mode combination** method.

The mode combination methods require different inputs. In particular, several of them provides a possibility to choose whether the coupling terms between modes are to be considered as always positive, or they may actually reduce the total response. This is controlled by the **Use absolute value for coupling terms** checkbox. Its default value differs between the methods, according to what is expected to be the most common choice.

Periodic Modes and Rigid Modes

For frequencies higher than the highest frequency content of the excitation, the SDOF system will respond as a rigid body. Some of the response spectrum evaluation methods take this into account. The effect is that high-frequency ("rigid") modes are assumed to have a higher degree of correlation than low-frequency ("periodic") modes. To take this effect into account, you can select **Rigid modes** to be the **Gupta** method or the **Lindley–Yow** method.

Missing Mass Correction

In general, only a small fraction of the total number of eigenmodes are used in the superposition. Therefore, some fraction of the total mass of the structure is not accounted for. The ignored modes usually have high natural frequencies. Such modes will not have a significant dynamic amplification. The mass in the corresponding SDOF system will just follow the movement of the foundation. This means that if the distribution of the missing mass is known, then it can be treated as an extra stationary mass force, where the peak acceleration during the event replaces the acceleration of gravity.

You can use two different methods for missing mass correction:

- In the **Missing mass method**, the difference between the true mass distribution and the mass represented by the used eigenmodes will act as extra static load. Typically, most of the missing mass is located close to support points, where the modal amplitudes are low. This method can be used together with either the **Gupta** method or the **Lindley–Yow** method.
- In the **Static ZPA method**, the total inertial force is used as static load. At the same time, only the periodic part of the response is used in the mode summation. This method can only be used together with the **Lindley–Yow** method because it is only compatible with the assumptions about how the rigid modes are scaled.

Static Solution

In some situations, you need to assess how the effects of a short event represented by a response spectrum solution interact with a static solution. A common case is when stresses from self-weight and service loads are significant.

The study providing the static loads is usually separate from the eigenvalue solution, and the two studies can be solved in any order. Any set of real-valued results can be used as the static load case. This includes results at a certain time step in a time-dependent study, and results for a certain parameter value in a stationary study with an auxiliary sweep.

To incorporate a static solution, select the **Add static solution** checkbox in the **Static Solution** section. When selected, you will be able to select a **Static solution** dataset in the **Data** section.

Response spectrum result are, per definition, positive. This means that any direction information is lost. When combining with another solution, an assumption about signs is necessary. In most cases, a worst case is obtained when the response spectrum result

is added using the same sign as in the static solution. How the solutions are summed is controlled by selecting an option from the **Sign of response spectrum solutions** list.

RESULT INTERPRETATION

All results from a response spectrum analysis are positive; the evaluation methods contain absolute values or RMS-like operations. This has important implications for the interpretation of the results:

- If the sign of your result quantity is important for the conclusions, you need to manually consider also the case of a negative value.
- It is not meaningful to evaluate a quantity that is not proportional to the mode amplitude. This does, however, not mean that you are limited to linear quantities; equivalent stresses like von Mises and Tresca are nonlinear, but they are still proportional to the mode amplitude.
- It is not meaningful to show deformation or vector plots of response spectrum results.

An underlying implicit assumption for response spectrum analysis is that it is performed in a frame of reference that follows the foundations of the structure. Thus, all displacements, velocities, and accelerations are relative to the foundation, which in itself is accelerating. Thus, it is difficult to evaluate the absolute acceleration. The conservative way of doing so is to add the peak acceleration of the underlying event to the computed acceleration. This will usually be rather conservative since the peak of the excitation seldom coincides in time with the peak of the response.

For displacements, the relative displacements between two objects within the same moving frame is usually a more important result than the absolute displacement, so it does seldom matter whether a constant value is added to all points or not.

- The theory is described in Response Spectrum Analysis Theory.
- The Response Spectrum node is described in the *COMSOL Multiphysics Reference Manual.*
- Combine Solutions and Updating a Solution in the *COMSOL Multiphysics Reference Manual.*
- The settings for the special datasets are described in Response Spectrum 2D and Response Spectrum 3D in the *COMSOL Multiphysics Reference Manual.*

	• *Earthquake Analysis of a Building*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/ building_response_spectrum** • *Shock Response of a Motherboard*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/ motherboard_shock_response**

Performing an Inertia Relief Analysis

Introduction

Inertia relief analysis is a type of stationary analysis, where the external loads are balanced by (unknown) inertial forces. Thus, even though rigid body motion constraints must be present to avoid a singular stiffness matrix, there will be no reaction forces.

The basic principle is that different load cases are studied. Arbitrary constraints, sufficient to suppress all rigid body motions (in 3D: six constraints), are added to the model. Then, six load cases (in 3D) are run for the six possible translational and rotational unit accelerations. Each such load case will give a set of reaction forces. A seventh load case, and corresponding reaction forces, are given by the external loads on the structure. It is now possible to scale the unit accelerations and sum them to the external loads so that the reactions are all zero.

For more information, see Inertia Relief in the *Structural Mechanics Theory* chapter.

Since the procedure is based on linear superposition of precomputed load cases, the inertia relief analysis can produce incorrect results in case of geometric nonlinearity.

Setting up this procedure manually is possible, but tedious and error prone. For this reason there is a built-in functionality for this.

Setting up an Inertia Relief Analysis

To perform an inertial relief analysis, you add an Inertia Relief node to the physics interface. You should not use any other types of constraints or elastic supports. The structure must be able to move freely as rigid body. The constraints needed to avoid singularity are added internally.

If you have several structural mechanics interfaces in your model, you should add an **Inertia Relief** node to each of them.

You can use any type of external loads, such as point loads, boundary loads, or forces on rigid objects.

In most cases, you do not need to change any settings in the **Inertia Relief** node. The constrained points will be automatically chosen by the program. You can, however, control which points that are used for the constraints. To do that, change **Control points** to **From selection input**. Some examples of cases when you may need to select the control points manually are:

- A point load is applied to a point.
- There are internal constraints (for example from multiphysics couplings) at a point.

When **From selection input** is selected, a **Point Selection** section is shown. Select the points to be used for constraining the domain.

- In 3D, select three points that are not collinear.
- In 2D, select two points.
- In 2D axisymmetry, select one point, not on the axis of rotation.

The next step is to prepare the model for the analysis. Click the **Create Load Groups and Study** button. If you have more than one physics interface, you should only click the **Create Load Groups and Study** button in one of them.

Do not manually edit the created load groups, or the settings for the load case handling in the study.

A special set of load groups will be added, together with a new study. This study consists of two stationary study steps. In the first step, a number of load cases (six in 3D) are solved. Using these load cases, a rigid body acceleration is determined, using the conditions that the external loads are balanced by inertia forces. In the second study step, the external load and the computed inertia forces are all combined together to form the total solution.

Once you have run the special training study for inertia relief, any other study will contain the computed rigid body acceleration field. If the external loads are changed, then you need to re-run the training study.

Bracket — Inertia Relief Analysis: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_inertia_relief**

Stress Linearization

The information about stress linearization is applicable if your license includes the Structural Mechanics Module.

Stress linearization is a procedure in which the stress distribution along a line through the thickness in a solid is approximated with an equivalent linear stress distribution, similar to what would be the result of an analysis using shell theory. The line is commonly referred to a *stress classification line,* SCL. This type of evaluation is common in the analysis of pressure vessels. It is also useful for the design of reinforcements for concrete structures and for fatigue analysis of welds.

To perform a stress linearization, you add one Stress Linearization node for each SCL.

ORIENTATION OF STRESS COMPONENTS

The stresses along the SCL are represented in a local orthonormal coordinate system, $x1$-$x2$-$x3$. The $x1$ direction is oriented along the SCL, so it is mainly the stresses in the second and third directions that are of interest.

- In 3D, you must specify the $x2$ direction and thus implicitly the $x3$ direction. You specify the orientation either by selecting a point in the $x1$-$x2$ plane or by defining an orientation vector in an approximate $x2$ direction. In either case, the actual $x2$ direction is chosen so that it is perpendicular to the SCL and lies in the plane you have specified. The $x3$ orientation is then taken as perpendicular to $x1$ and $x2$. As long as you are only interested in a stress intensity, the choice of orientation is arbitrary.
- In 2D, the $x3$ direction is the out-of-plane direction, and the $x2$ direction is perpendicular to the SCL in the XY-plane.
- In 2D axial symmetry, the $x3$ direction is the azimuthal direction, and the $x2$ direction is perpendicular to the SCL in the RZ-plane.

CREATING THE STRESS CLASSIFICATION LINE

The most straightforward way to create an SCL is to include it in the geometry, and then select it in a **Stress Linearization** node. This corresponds to using the **Line** linearization type where the SCL is defined by **Edge**.

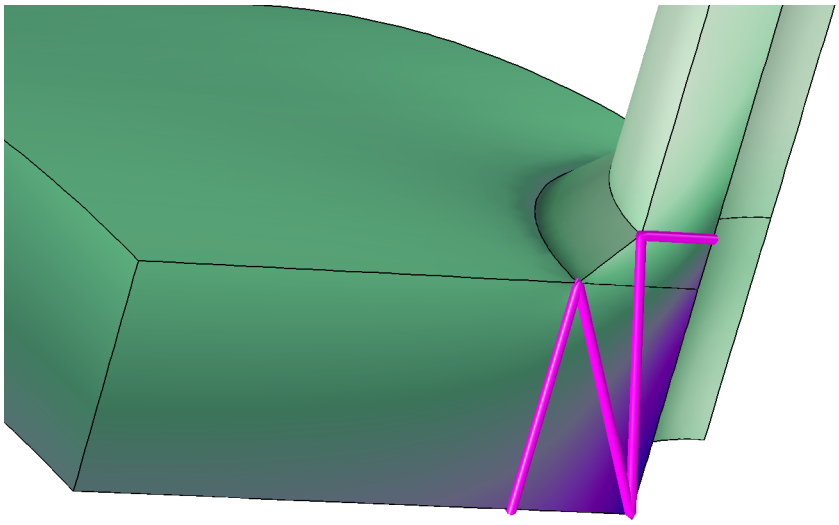


Figure 2-49: Four stress classification lines in a transition region at a pressure vessel nozzle.

There are, however, some situations where this direct approach is less convenient:

- When the introduction of the SCL in the geometry must be done after the analysis, since the locations of critical regions were not obvious when setting up the initial analysis. It is of course possible to add new edges and rerun the analysis, but this may not be a good solution if the analysis time is long.
- When the introduction of the edges for the SCL makes the meshing more difficult. It may, for example, not be possible to use swept meshes anymore, or the mesh quality is reduced in critical regions.

In the two above cases an alternative is to define an SCL that is not included in the geometry by connecting two arbitrary points with a straight line. This corresponds to using the **Line** linearization type where the SCL is defined by **Two points**. The introduction of the SCL will in this case not make it necessary to update the geometry and rerun the analysis. Since the SCL is disconnected from the geometry and mesh, this option will also make meshing easier. The downside is then the you cannot control the mesh quality along the SCL.

In some cases it is difficult to *a priori* determine where to create the SCL or the critical location may change depending on, for example, loading conditions or time. For such cases, it is also possible to define it in a distributed manner such that an infinite number of SCLs are created in a domain. The starting point of the SCLs are given by a boundary selection, and they extend along the normals of the boundary through the selected domains. This corresponds to using the **Distributed** linearization type.

When you do a distributed evaluation, and find a critical point using a surface plot over the boundary, you will typically want to insert an SCL from that point. You can do that in the following manner:

1. Add a **Marker** subnode to the **Surface Plot**.
2. In the **Graphics** window, click on the maximum value marker. This automatically populates the **Evaluation 3D** table with the coordinates and the value of the plotted expression.
3. In the **Evaluation 3D** table, right-click on the newly added values and select **Copy Selection to Clipboard**.
4. Go to the **Stress Linearization**, and expand the **Results** section.
5. Select the **Linearization line, starting point** table and press CTRL+V. This inserts the coordinates into the table.
6. You can now update the solution, and get access to the results for the newly defined SCL.

STUDIES AND SOLUTIONS

Stress linearization is a pure postprocessing operation. The **Stress Linearization** node will only create a number of variables, which can be evaluated under **Results**. It is thus possible to add such nodes after the main analysis has been performed. In order to make the new variables available for postprocessing, you must then run an **Update Solution**.

RESULTS

When you have included one or more **Stress Linearization** nodes in a model, a number of datasets and an extra result template are generated.

One edge dataset is created for each SCL. These datasets are named `Linearization Line <`*n*`>`, where *n* is an integer number.

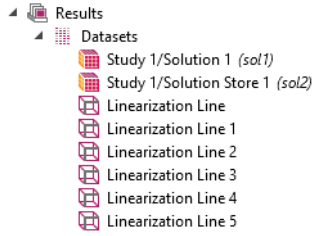


Figure 2-50: Generated datasets in a model with five SCL.

The plot contains graphs for the 22 component of the actual stress, the membrane stress, and the linearized stress. The first **Linearization Line** dataset is selected. By changing edge dataset in the plot group, you can easily move between the different stress classification lines.

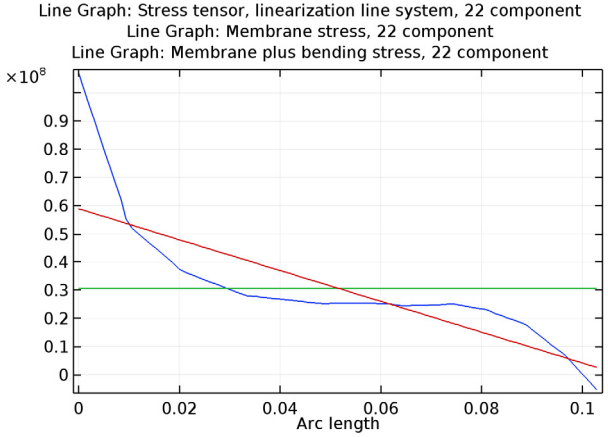


Figure 2-51: Plot along a stress classification line.

To switch to another stress component, you will have to edit **Expression** fields in five different places per plot. A quick way of doing that is from the **Find and Replace** window (Ctrl+F). In that window, do the following:

- Enter the current stress component (for example 22) in the **Contains** text field.
- Enter the desired stress component (for example 33) in the **Replace with** text field.
- From the **Add Filter** list, select **Settings**.

- In the **Settings**. dialog box, go to the **Name** text field, and enter the string `expression`.
- Select the **Contains word** checkbox, and click **OK**.
- When you click **Search**, you get a list of matching strings, and can select the ones to replace.

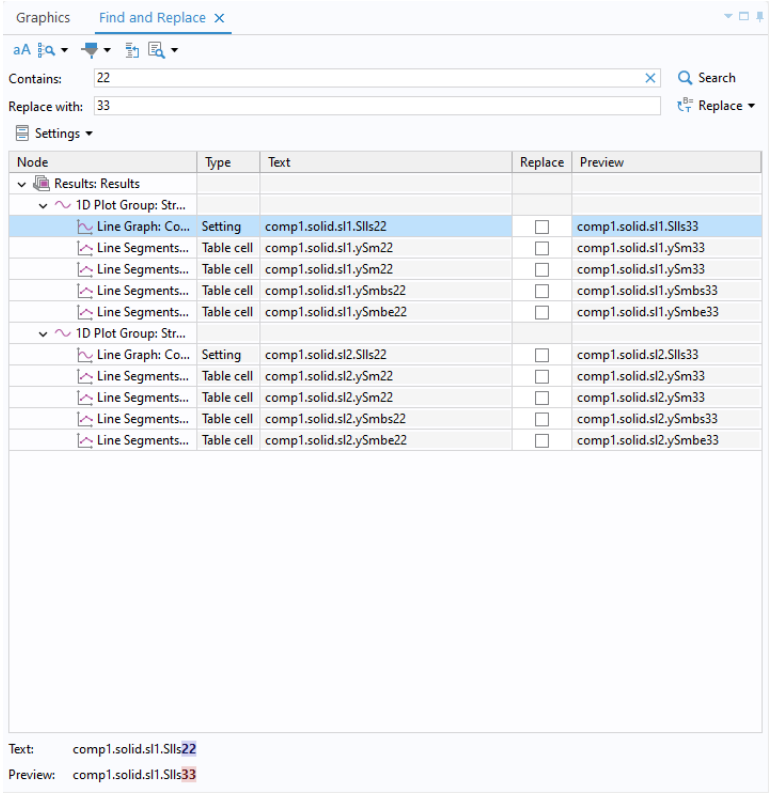


Figure 2-52: The Find and Replace window.

For the **Distributed** linearization type, the created **Linearization Line** dataset corresponds to a single SCL in selected domain. The starting point is specified in the **Postprocessing** section of the feature, and must lie on the selected boundaries. The endpoint is computed internally such that the cut line extends through the selected domains along the boundary normal of the point.

VARIABLE NAMES

Each **Stress Linearization** node adds a number of variables. Many of these variables exist with two different scopes, *physics scope* and *feature scope*. The physics scope is the name of the physics interface, having the default value '`solid`'. The feature scope contains also the tag of the **Stress Linearization** node, for example, '`sl1`'.

As an example, the variable `solid.Sm22` and the variable `solid.sl1.Sm22` have the same value. The variables with physics scope make it more convenient to create expressions using postprocessing. You could, for example, make a line plot of `solid.sb22` and get all edges having a stress linearization colored by their individual results.

TABLE 2-21: VARIABLES FOR STRESS LINEARIZATION.

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Sllsij	Stress tensor in local coordinate system	σ_{ij}	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Smij	Membrane stress	$\sigma_{m,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Sbmaxij	Maximum bending stress	$\sigma_{b(max),ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
Sbij	Bending Stress	$\sigma_{b,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Smbij	Membrane + bending stress	$\sigma_{mb,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Spsij	Peak stress, starting point	$\sigma_{p(start),ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
Speij	Peak stress, starting point	$\sigma_{b(end),ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
Slm	Stress intensity, membrane	σ_{int}		Physics, Feature
Slmbs	Stress intensity, membrane + bending, starting point	σ_{int}		Feature
Slmbe	Stress intensity, membrane + bending, endpoint	σ_{int}		Feature

TABLE 2-21: VARIABLES FOR STRESS LINEARIZATION.

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
SImb	Stress intensity, membrane + bending,		max(SImbs, SImbe)	Physics, Feature
mises_m	Equivalent von Mises stress (membrane)			Physics, Feature
mises_b	Equivalent von Mises stress (membrane)			Physics, Feature
mises_mb	Equivalent von Mises stress (membrane plus bending)			Physics, Feature
tresca_m	Equivalent von Tresca stress (membrane)			Physics, Feature
tresca_b	Equivalent von Tresca stress (membrane)			Physics, Feature
tresca_mb	Equivalent von Tresca stress (membrane plus bending)			Physics, Feature
N*ij*	Local in-plane force	N_{ij}	*ij* = 22, 23, 33	Feature
M*ij*	Local bending moment	M_{ij}	*ij* = 22, 23, 33	Feature
Q*i*	Local out-of-plane shear force		*i* = 2, 3	Feature
lengthtot	Length of SCL	L		Feature
arclength	Coordinate along SCL	X_1		Feature

Stress Linearization in the *Structural Mechanics Theory* chapter.

Solver Settings for Structural Mechanics

COMSOL Multiphysics includes many solvers and solver settings. To make it easier to use a suitable solver and its associated solver parameters, the physics interfaces have different default settings based on the study type and features used. In some situations, the default settings may need to be changed. This section helps you to select a solver and its settings to solve structural mechanics and related multiphysics problems.

In this section:

- Symmetric Matrices
- Selecting Iterative Solvers
- Specifying Tolerances and Scaling for the Dependent Variables

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

If you make changes to the physics, this will not be reflected in the solver settings unless you regenerate the solver sequence.

Symmetric Matrices

Use the **Matrix symmetry** list (see the **General** section in the **Settings** window for **Advanced** for a solver node such as **Stationary Solver**). There you can explicitly state whether the assembled matrices (stiffness matrix, mass matrix) resulting from the compiled equations are symmetric or not.

Normally the matrices from a single-physics structural mechanics problem are symmetric, but there are exceptions, including the following cases:

- Multiphysics models solving for several physics simultaneously (for example, heat transfer and structural mechanics). Solving for several Structural Mechanics interfaces, such as shells combined with beams, does not create unsymmetrical matrices.

- Contact with friction.
- Elastoplastic analysis

Advanced and Stationary Solver in the *COMSOL Multiphysics Reference Manual*

One of the benefits of using the symmetric solvers is that they use less memory and are faster. The default option is **Automatic**, which means the solver automatically detects if the system is symmetric or not. Some solvers do not support symmetric matrices and always solve the full system regardless of symmetry.

Selecting the **Symmetric** option for a model with unsymmetric matrices may lead to incorrect results. For a nonlinear problem that is only weakly unsymmetric, it may still be useful, since the faster solution of the symmetric problem may offset the lower convergence rate. This is, for example, the case for contact problems with a low coefficient of friction.

Complex matrices can be unsymmetric, symmetric, or Hermitian. Hermitian matrices do not appear in structural mechanics problems.

Selecting the **Hermitian** option for a model with complex-valued symmetric matrices produces incorrect results.

Selecting Iterative Solvers

The default solver for structural mechanics is the MUMPS direct solver in 2D and the PARDISO direct solver in 3D. For large 3D problems (several hundred thousand or millions of degrees of freedom) it is beneficial to use iterative solvers when possible to save time and memory. The drawback is that they are more sensitive and might not converge if the mesh quality is low. The iterative solvers also have more options than the direct solvers.

For Stationary and Time Dependent studies and Frequency Domain studies in 3D, a GMRES iterative solver is preconfigured and available as an alternative solver suggestion for solid mechanics models.

If the model is set up using quadratic (default) or higher-order elements for the displacement dependent variable, this GMRES solver will use geometric multigrid (GMG) as the preconditioner.

For slender geometries, changing to SOR Line as presmoother and postsmoother can give better results compared to SOR, which is the default for the GMG preconditioner.

For models using linear elements for the displacement dependent variable, the preconditioner will be changed to smoothed aggregated algebraic multigrid (AMG). This is to avoid remeshing when creating the discretization on the coarse level. Note that you need to manually regenerate the solver sequence after you change the element order if you want to make use of such change in the predefined iterative solver configuration.

For eigenfrequency and eigenvalue studies, use the default direct solver

Specifying a shift frequency greater than the lowest eigenfrequency results in indefinite matrices.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

Solver Settings for Viscoelasticity, Creep, and Viscoplasticity

Viscoelasticity, creep, and viscoplasticity are time-dependent phenomena. The time scale, however, is often such that inertial effects can be ignored. When that is the case, you can modify the solver settings to improve the performance of the time-dependent analysis.

PHYSICS INTERFACE SETTINGS

In the **Model Builder**, click the **Solid Mechanics** node. In the **Settings** window, under **Structural Transient Behavior**, select **Quasi static** to treat the elastic behavior as quasi static (with no mass effects; that is, no second-order time derivatives for the displacement variables). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system since no mass matrix will be created.

SOLVER SETTINGS

When **Quasi static** is selected on the physics interface **Settings** window, the automatic solver suggestion changes the method for the **Time Stepping** from **Generalized alpha** to **BDF**.

For a **Fully Coupled** node (or **Segregated** node for multiphysics problems), the default **Nonlinear method** under **Method and Termination** is **Automatic (Newton)**. To get a faster computation time when the strain rate is low or moderate, select **Constant (Newton)** as the **Nonlinear method** instead.

In the *COMSOL Multiphysics Reference Manual:*

- Studies and Solvers
- Fully Coupled and Segregated

Specifying Tolerances and Scaling for the Dependent Variables

The absolute tolerance parameters used for time-dependent studies are problem specific. By default, the absolute tolerance is applied to scaled variables, with the default value being 0.001 for all solution components.

The default scaling for the displacement components is based on the size of the geometry in the model, and certain reasonable scales are used for the pressure and contact force variables, if any. You are encouraged to change these scales as soon as better values are known or can be guessed or estimated from the applied forces, yield stress, reaction forces, maximum von Mises stress, or similar. The same suggestion applies to the displacement scale, which can be estimated easily if the problem is displacement controlled. This approach can significantly improve the robustness of the solution. The variable scaling is accessed under **Dependent Variables** in the solver sequence. The scales need to be entered using the main unit system in the model.

Solver Settings

In many situations, the default in COMSOL Multiphysics when having several physics interfaces is to generate a solver sequence with a segregated solver. When several structural mechanics interfaces are present, it can happen that the degrees of freedom are placed in different segregated steps by the default solver generation. It is, however, not possible to solve a model where the structural mechanics degrees of freedom are

placed in different segregated groups, so in this case you must modify the solver settings.

- If the model only consists of structural mechanics interfaces, the best option is usually to replace the segregated solver with a fully coupled solver.
- If the model contains contact conditions where the augmented Lagrangian method is used, there are two versions of the algorithm: one based on a special segregated solver, and one based on a fully coupled solver. In the first case, a segregated solver must be used. Place all structural mechanics degrees of freedom except the contact variables in one segregated step. The contact variables should remain in the lumped step, which should appear directly after the step with the other mechanical degrees of freedom.
- If there are other types of physics interfaces than structural mechanics being solved in the same study, then the segregated solver should usually be kept. Make sure that all structural mechanics degrees of freedom except segregated contact variables are solved in one segregated step.

In the *COMSOL Multiphysics Reference Manual*:

- Solution Operation Nodes and Solvers
- About the Stationary Solver

Result Presentation

In this section, some special postprocessing techniques that are particularly useful for structural mechanics models are presented.

In this section:

- Local Orientations
- How an Expression Is Evaluated
- Quality Settings
- Gauss-Point Evaluation
- Postprocessing of Eigenmodes
- Plotting Applied Loads

General principles for result presentation and how to work with the features under **Results** in the Model Builder are provided in Results Analysis and Plots in the *COMSOL Multiphysics Reference Manual*.

Local Orientations

Many stress and strain tensor components are available both in local and global directions. Some examples are:

- `solid.sxy` — Cauchy stress with respect to the global spatial coordinate system
- `solid.sl12` — Cauchy stress with respect to the local coordinate system of the material.
- `solid.SXY` — Second Piola–Kirchhoff stress with respect to the global material coordinate system
- `solid.Sl12` — Second Piola–Kirchhoff stress with respect to the local coordinate system of the material
- `solid.eXY` — Strain with respect to the global material coordinate system
- `solid.el12` — the local coordinate system of the material

As can be seen, tensor component indices containing the names of a coordinate directions has orientations along those axes. Components in local directions contain the letter 'l' and numerical indices.

The local directions are defined by the coordinate system attached to a material model. For most material models, you can select a local coordinate system in the **Coordinate System Selection** section of its settings. The purpose of this selection is twofold:

- It provides the orientations along which orthotropic and anisotropic material data are interpreted.
- It acts as the coordinate system in which the local stress and strain tensor components are presented.

When the material is isotropic, you can utilize the local coordinate system just for the second purpose. In this case, the results of the analysis itself are independent of the selected coordinate system. It just provides a transformation to be utilized during result evaluation. If you want to change the orientations for the local components of the tensors, you do not have to solve the study again. Instead, you select the new coordinate system in the material model node, and then run **Update Solution** ( ) for the study or studies where you want the new definition to be applied.

Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

ADDING YOUR OWN TRANSFORMATIONS

If there are no suitable local coordinate system variables defined from the physics interface itself, you can create your own transformations.

The easiest way of doing that is to use the Local System Results node, available in some of the structural mechanics interfaces. Here, the input is just a local coordinate system.

A more general approach is to add a Vector Transform or Matrix Transform under **Definitions** > **Variable Utilities**, depending on the type of object you are going to transform.

These nodes provide a large degree of flexibility in defining various types of transforms, but for a pure rotation into a new coordinate system, the settings are straightforward. Do the following:

1 Add the local coordinate system in which the results are to be presented under **Definitions**.

2 In the **Input** section, enter the components of the vector or tensor to be transformed. In most cases, you can pick it using **Replace Expression** ( ).

	Some tensors within the structural mechanics interfaces do not contain an intrinsic information about the associated coordinate system. This is in particular true for those tensors labeled as 'local'. When using them in the **Matrix Transform** node, you must then manually select the coordinate system for the input tensor.

3 In the **Vector Transform** or **Matrix Transform** node, select the local coordinate system in the **Output** section. In the latter node, select it twice.

4 Select the domain where you want this transformation to be valid.

5 Often, you want to enter your own name for the transformed object in the **Name** text field.

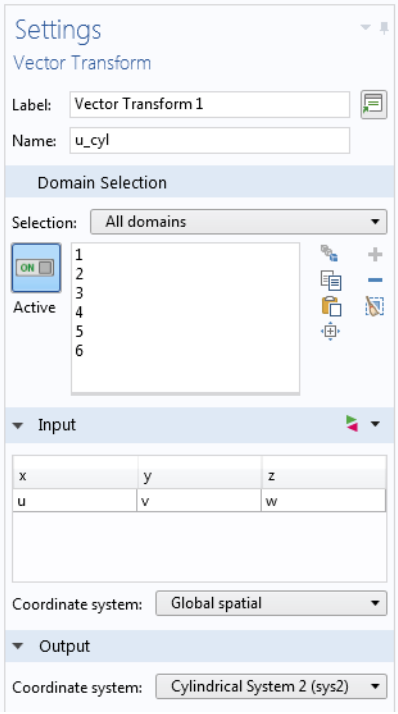


Figure 2-53: Example of settings for transforming the displacement vector to a cylindrical coordinate system.

If you define the transformation after the study was solved, you will need to run **Update Solution** ( ) for the study or studies where you want to access the transformed variables.

You can now access the new transformed quantities for various type of result presentation. They are accessible under **Definitions** in the **Replace Expression** dialog.

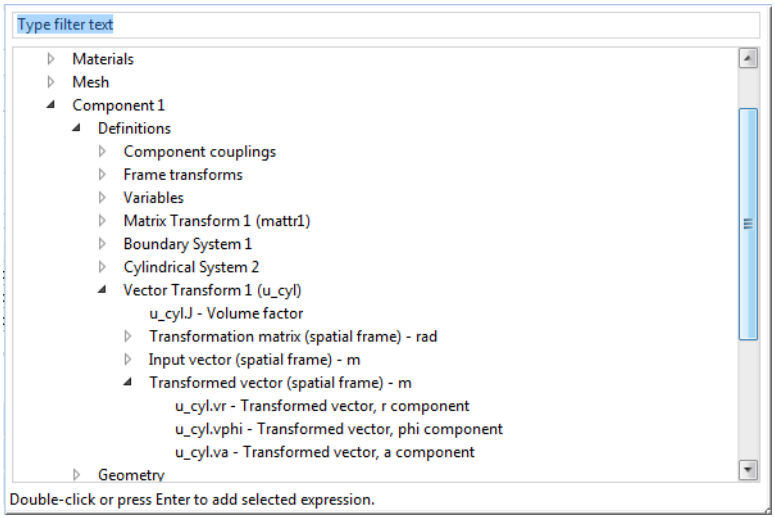


Figure 2-54: Picking a transformed result in the Replace Expression dialog.

	Vector Transform and Matrix Transform in the *COMSOL Multiphysics Reference Manual.*

How an Expression Is Evaluated

In order to get optimal quality in the result presentation, it is necessary to have some understanding about how expressions are evaluated.

In structural engineering, the maybe most commonly evaluated result quantities are stresses and strains. Stress expressions can, however, be sensitive to specific settings during result presentation, and are for this reason used as an example in the discussion below.

Consider a linear elastic material with thermal expansion and creep. The stress tensor σ is then computed as

$$\sigma = \sigma_{ex} + \mathsf{C} : \varepsilon_{el} = \sigma_{ex} + \mathsf{C} : (\varepsilon - \varepsilon_{inel})$$

where C is the 4th-order elasticity tensor, ":" stands for the double-dot tensor product (double contraction). The elastic strain ε_{el} is the difference between the total strain ε and all inelastic strains ε_{inel}. In this example, the inelastic strain tensor has two contributions, creep strain ε_{cr} and thermal strain ε_{th}. There may also be an extra stress term σ_{ex} with contributions from initial stress and external stress.

When a stress tensor component is to be evaluated, the following happens:

- A position $\mathbf{x}$ in an element is identified, where the value is to be computed.
- The total strain ε is computed from the derivatives of the shape functions at $\mathbf{x}$. The strain depends on the values of the displacement degrees of freedom at the nodes and on the shape functions in the element.
- The expression giving the thermal strain ε_{th} is evaluated at $\mathbf{x}$. This expression will depend on a temperature, which may be prescribed or computed in another physics interface.
- The creep strain ε_{cr} is a state variable, which is stored at the Gauss points in the element. The value is picked from the Gauss point closest to $\mathbf{x}$.
- The inelastic strain ε_{inel} (in this example, the creep strains is now subtracted from the total strains, to form the elastic strain ε_{el}.
- The elastic strain ε_{el} is multiplied by the elasticity tensor C, evaluated at $\mathbf{x}$, to give the stress σ. The material properties may depend on the location, either explicitly, or for example by a temperature dependency.
- Any extra stress σ_{ex}, evaluated at $\mathbf{x}$, is added to the stress tensor.

The subtraction between total and inelastic strains is however a sensitive operation. It is not uncommon that these two terms are close to each other in value, so that the resulting elastic strain is a small number obtained by subtraction of two larger numbers. Since the three types of strain in this example have different types of distribution through the element, there is a risk that this difference will exhibit artificial variations inside the element.

As an example, let's assume that the temperature has a quadratic variation through the element, and that standard quadratic shape functions are used in Solid Mechanics. The total strain is a linear function within each element, since it contains derivatives of the shape functions. The computed elastic strain is the difference between the linear total

strain and the quadratic thermal strain. If there also are creep strains present, they will be piecewise constant, since they are taken from the nearest integration point.

It must, moreover, be realized that the total strain is 'correct' only in an average sense. It provides a kind of best fit given by the finite element formulation.

Quality Settings

In many features under **Results**, such as surface plots and line graphs, there is a section named **Quality**, through which details of how the results are computed can be specified. In some situations, you need to modify these settings to obtain optimal accuracy or visually attractive presentations.

First, it can be noticed that stresses, strains, and any other variables that are based on derivatives of the shape functions are not continuous over element borders. The **Smoothing** control is used for improving plots by applying an averaging between neighboring elements. The default when you add a result feature is to use smoothing, and this will in general improve the results. It does however also hide any jumps in the solution, which could indicate an inadequately resolved mesh. As soon as you are investigating the quality of the solution, it is a recommended to avoid smoothing. In the default stress plots that are generated by the structural mechanics interfaces, an intermediate path is taken: the **Smoothing threshold** is set to **Manual**, with the **Threshold** value set to 0.2. This means that as long as the values from adjacent elements do not differ by more than 20%, smoothing is applied. If, however, there are significant jumps in the solution, they will be clearly visible.

The **Resolution** setting determines at how many points in the element the result quantity is evaluated. Essentially, a local finer mesh is used within the element for the visualization. Using a high resolution can be problematic and lead to local overshoots and artificial 'waviness' of the solution. For smooth expressions, like a stress in the absence of inelastic strains or a displacement, this is not a problem. If, however, the function has sudden variations within the element, a high resolution will give results having artificial variations.

The default resolution depends on several factors, but will generally result in that a result quantity is evaluated in a rather large number of points in the element. In addition to affecting plot quality, having a high resolution can make the plotting significantly slower.

Using a low resolution will remove the sudden variations, but may still not be an optimal choice. If you set the quality to **No refinement**, the expression is evaluated only

at the corners of the element. This is, for example, rather far from the Gauss points where many types of inelastic strains are stored. Also, a thermal strain with a quadratic distribution will be rather far from its best linear approximation at the element corners.

The most robust solution if you encounter this type of problems it to create a low-order field that approximates the "wavy" expression over the element, as described in the next section.

Entering Quality Settings for Plots and Plot Groups in the *COMSOL Multiphysics Reference Manual.*

Gauss-Point Evaluation

You can create a smooth field within each element for any expression by using the `gpeval()` operator. In its most basic version, this operator has the following syntax: `gpeval(integration_order, expression)`.

For a general description of the `gpeval()` operator, see gpeval in the *COMSOL Multiphysics Reference Manual.*

The `gpeval()` operator evaluates an expression at a given set of Gauss points and creates an approximate smooth field using the least squares fit. It is important to use the correct integration order to accurately evaluate state variables stored at Gauss points, as in the case with many nonlinear material models. For example, when quadratic shape functions are used with full integration, the first argument to the `gpeval()` operator should be 4.

The physics interfaces define built-in variables with Gauss-point evaluation for many variables, including stress and strain. As an example, the built-in variable `solid.sGpxx` is equivalent to using the expression `gpeval(4,solid.sxx)`. Another example is `solid.misesGp`. However, similarly using `gpeval(4,solid.mises)` can lead to unphysical negative values caused by the extrapolation done by the operator. Hence, for a variable that is strictly positive, one should use

`max(0,gpeval(4,solid.mises))` or similar expressions; this is done for all relevant built-in variables with Gauss-point evaluation.

When adding predefined quantities (**Add/Insert/Replace Expression** or Ctrl+Space) to an expression field, some variables including stress and strain are presented with different definitions depending on where in the model tree you are working.

- When modeling in the physics interfaces, or generally under **Definitions**, the standard definition of variables is presented.
- When working under **Results**, the Gauss-point evaluation definition of variables is presented.

The variables that are visible in a certain context can be controlled using the **Show All Variables** checkboxes in the **Show More Options** ( ) dialog.

When working with surface plots in a 3D geometry, or with line graphs in 2D and 3D, there is a subtle difference between the built-in variables and the operator syntax. The built-in variables are defined in the domain, and the field is projected from the domain to be evaluated on boundaries and edges. If, however, you type in the operator syntax as an expression in a line graph, the field is created only on that line, using only Gauss points on the element edges. In order to avoid this, create domain variables under **Definitions** > **Variables** using the operator syntax, and then use them on boundaries or edges. Another alternative is to use the `mean()` operator, which evaluates any expression in adjacent entities of higher order. For example, to evaluate a domain variable in different plots in 3D, use the following syntax:

- Surface plot: `mean(gpeval(integration_order, expr))`
- Line graph: `mean(mean(gpeval(integration_order, expr)))`
- Point graph: `mean(mean(mean(gpeval(integration_order, expr))))`

There are also built-in operators that can be used specifically for evaluating domain variables in any type of plot. Their syntax is summarized in Table 2-22. These operators also automatically set the integration order to be consistent with its parent physics

interface <phys>, as opposed to the corresponding general operators. Also, for example, reduced integration is taken into account.

TABLE 2-22: SUMMARY OF OPERATORS FOR EVALUATION EXPRESSION IN GAUSS POINTS OF DOMAIN MESH ELEMENTS.

OPERATOR SYNTAX	EXPLANATION
<phys>.elemavg(expr)	Elementwise average in domain mesh elements
<phys>.elemgpmax(expr)	Elementwise maximum over Gauss points in domain mesh elements
<phys>.elemgpmin(expr)	Elementwise minimum over Gauss points in domain mesh elements
<phys>.elemint(expr)	Elementwise integration in domain mesh elements
<phys>.gpeval(expr)	Elementwise Gauss-point evaluation in domain mesh elements

Postprocessing of Eigenmodes

EIGENMODE SCALING

The eigenmodes that form the solution to eigenvalue problems, like eigenfrequency and linear buckling, have some special properties that require attention. The most important property is that an eigenmode is only defined up to an arbitrary multiplicative factor. Thus, the actual values of the modal displacements have no physical significance. In order to emphasize this, the default eigenmode plots have no color legend.

The underlying theory does, however, assume that the mode shape is an infinitesimal perturbation to the geometric shape.

If the eigenmodes have an unfortunate scaling, confusing effects can appear during postprocessing:

- Any variable or expression used during postprocessing of an eigenvalue solution will be evaluated using the setting of the **Include geometric nonlinearity** checkbox of the eigenfrequency study step. If the values of the modal displacements are large and the study step is geometrically nonlinear, then the nonlinear parts of the strain tensor may become large. This is not consistent with the assumption of a small perturbation.

- If the study step is geometrically linear, there can be other types if inconsistencies if the modal displacements are large. Many variables are then defined under the assumption that angles are small. As an example, the normal vectors to a shell can appear to no longer have unit length.
- Rotational degrees of freedom scale the same way as the displacements. Computed rotations can potentially get values of more than a full revolution.

In most cases, you only have to be aware of these phenomena. But if you really need to access quantitative values (for example, as modal stresses), you need to use some caution.

In a response spectrum analysis, all result quantities are computed by scaling the modal results. In this case, it is necessary that all eigenmodes are scaled so that the assumptions of geometric linearity are fulfilled.

The scaling of eigenmodes can be controlled in the **Output** section in the settings for the Eigenvalue Solver node. By setting **Scaling of Eigenvectors** to **Maximum** and using a small value for the **Maximum absolute value**, you can force the eigenmodes to be small. All structural mechanics interfaces override the default value and set it to 10^{-6} times the size of the bounding box of the geometry in order to keep the eigenmode displacements small.

DEFORMATION PLOT ARTIFACTS

The automatic scaling of a deformation plot can give strange impressions, in particular when the main deformation shape is a rotation. The geometry then seems to become

wider, since the displacements are directed in the infinitesimal tangential direction rather than along a circular path.

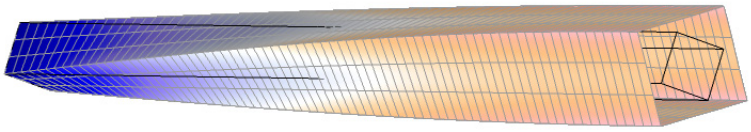


Figure 2-55: A torsional eigenmode in a box beam.

EIGENMODES IN A DEFORMED STRUCTURE

When you perform a prestressed eigenvalue analysis, it is possible that the prestress load case causes a significant deformation to the structure. The default eigenmode plot will still show the mode relative to the original undeformed structure. Sometimes this does not give a good enough representation of actual mode shape.

To improve the visualization, do the following:

1. Go to the settings for the 2D or 3D plot group containing the mode shape plot. Under **Plot dataset edges**, set **Frame** to **Spatial**. This will make the outline to be given by the deformed shape from the prestress load case.
2. In the dataset for the eigenvalue solution, set **Frame** to **Spatial**. The mode shapes are now plotted relative to the deformed shape.

Plotting Applied Loads

When solving, result templates containing the applied loads are generated. You can add them from the **Result Templates** window. Figure 2-56 shows an example of such a plot.

In this section, it is described in detail how you can work with these plots to fine-tune the visualization.

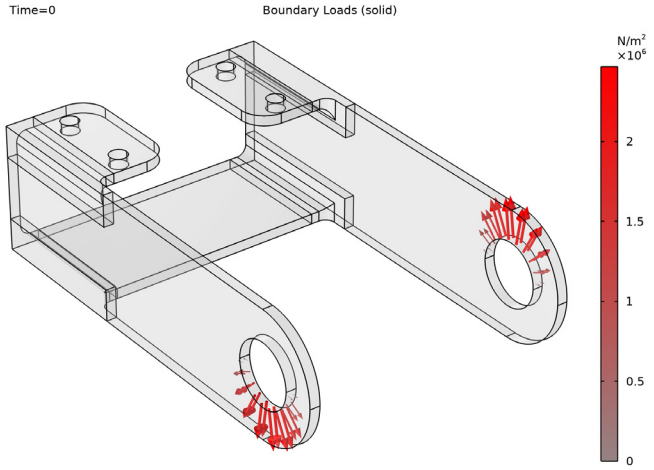


Figure 2-56: Example of an automatically generated plot to visualize the boundary loads applied to a bracket.

STRUCTURE OF DEFAULT LOAD PLOTS

For each structural mechanics physics interface and dataset, a node group is generated under **Results**. Its name is **Applied Loads (<interfaceTag>)**. Within this node group, one plot group per type of load is created. The possible plot group names are listed in Table 2-23.

TABLE 2-23: PLOT GROUP NAMES AND PLOT TYPES.

LOAD GROUP NAME	PLOT TYPE	DESCRIPTION
Volume Loads	Arrow Volume, Arrow Surface, Arrow Line, Arrow Point	Generated from **Body Load**, **Gravity**, **Linearly Accelerated Frame**, and **Rotating Frame**. The plot type depends on the geometrical dimension of the physics interface.
Boundary Loads	Arrow Surface	Generated from **Boundary Load**. Also, from **Added Mass** on a boundary combined with **Gravity** or **Rotating Frame**.
Face Loads	Arrow Surface	Generated from **Face Load** in Shell, Plate, and Membrane interfaces.
Face Moments	Arrow Surface	Generated from **Face Load** in Shell, Plate, and Membrane interfaces of nonzero moments are present.

TABLE 2-23: PLOT GROUP NAMES AND PLOT TYPES.

LOAD GROUP NAME	PLOT TYPE	DESCRIPTION
Edge Loads	Arrow Line	Generated from **Edge Load**. Also, from **Added Mass** on an edge combined with **Gravity** or **Rotating Frame**.
Edge Moments	Arrow Line	Generated from **Edge Load** in Shell, Plate, and Beam interfaces if nonzero moments are present.
Point Loads	Arrow Point	Generated from **Point Load**.
Point Moments	Arrow Point	Generated from **Point Load** in Shell, Plate, and Beam interfaces if nonzero moments are present.
Global Loads, Global Moments	Point Trajectories	Generated from global features like **Rigid Connector** and **Rigid Material**, which have subnodes as **Applied Load** and **Applied Moment**. Such loads are a type of point load, but they are applied at a certain location in space, rather than at a point in the geometry.
Fluid Loads (Pressure)	Line	Generated from **Fluid Load** in Pipe Mechanics interface.
Fluid Loads (Drag Force)	Arrow Line	Generated from **Fluid Load** in Pipe Mechanics interface.

In general, load groups are generated as soon as there is load of a certain type, irrespective of whether the value of the load is nonzero or not. Load features in which it is possible to enter both a force and a moment contribution, such as **Edge Load** in the Beam interface, have a special behavior:

- A plot group for forces is always generated.
- A plot group for moments is only generated if there is a nonzero value in at least one of the components of the moment vector.

Within a load group, there is one plot per load feature. In most cases, the plot has the same label as the generating load feature, for example **Boundary Load 2**.

MODIFYING THE VISUALIZATION

The default appearance of the load plots provides a good overall visualization of the loads applied to the model. However, for complex models or for creating plots for export, you might want to modify the default setting. Most default settings are presented in the following together with some prepared alternative settings and suggestions that you can make use of. While the plots are automatically generated, you

can modify them in any way you want to improve the visualization for your specific model; the plots are not reset each time you compute the study.

Style Inheritance

Within a load group, the style of each load plot is inherited from the previous plot. The inheritance can be modified in the **Inherit Style** section. For example, clear the **Color** checkbox to enable different color schemes for different loads and the **Color and data range** to allow different ranges of the color legends.

Filled Surfaces

In order not to hide any load arrows, the default is to use a transparent representation of the geometry. This is obtained through the extra surface plot **Gray Surfaces** together with a **Transparency** subnode. This plot adds a uniform gray surface on each boundary of the geometry. The effect is shown in Figure 2-56. If you disable the **Gray Surfaces** plot, the structure will have a wireframe outline only. You can modify the transparency level in the **Transparency** subnode, or delete the node completely. You can, furthermore, add a **Selection** subnode to the **Gray Surfaces** node to modify on which boundaries to plot the gray surface.

The **Gray Surfaces** plot is only generated for physics that use a 2D or 3D representation of the geometry.

Arrow Heads

You can change the attachment of the arrow by changing the setting of **Arrow base** in the **Coloring and Style** section. The default is in most cases to place the tail on the geometry. One exception is, for example, a pressure boundary load for which the head of the arrow is placed at the geometry.

Number of Arrows

For distributed fields, the default is to place one arrow at the center of each element, element face, or element edge. If the density of arrows is not appropriate, you can modify the settings in the **Arrow Positioning** section. For example, change **Placement** to **Mesh nodes** to place an arrow in each point of the linear mesh, or set it to **Uniform** to specify an arbitrary number of arrows to be plotted.

Color Scheme

The color scheme of the arrow plots is controlled in a **Color Expression** subnode. The default color table is a **Gradient** going from red to black. You can change this in the **Coloring** list. By selecting **Color table**, the color table is automatically set to **Spectrum**. This color table can be useful for a clear visualization of the applied loads, when, for example, the magnitude varies significantly between load features.

Deformation

By default, the load plots are presented in the initial configuration of the model. However, when applicable, a **Deformation** subnode is added to each plot, but with the **Scale factor** set to zero. Modify the scale factor to view the loads in a deformed state. Note that when multiple load plots exist in a plot group, the scale factor is by default inherited from the first plot. The exception is the **Gray Surfaces** plot, where the scale factor is set independently of the other plots.

When the analysis includes geometric nonlinearities, the **Scale factor** is by default set to one. Hence, the loads are presented in the deformed configuration for such cases.

- Under geometric nonlinearity, the size of the boundary load arrows will scale with the local area deformation. Thus, a constant pressure may appear as nonconstant.
- Load plots that make use of a **Point Trajectories** node include the displacements explicitly in order to visualize the current position of the load.

USING LOAD PLOTS DURING PREPROCESSING

The automatically generated plots of the applied loads can be useful during preprocessing, for example to verify that loads are applied on the correct boundary and in the intended direction. To generate the load plots before computing the study, right-click the **Study** node, and select **Get Initial Value**, see Computing the Initial Values. Doing so will generate all default plots and result templates requested by the currently active features in all physics interfaces active in the study. You can then insert the plot from the **Add Plot** window.

Note that plots are only generated the first time a dataset is created, which means that the settings will not be updated for subsequent computation of the solution. Moreover, load plots will not be generated for load features that are added after the creation of the dataset. To generate new plots when new load features have been added, delete the current load plots. Then compute the solution or do the **Get Initial Value**, and insert the new load plots from the **Add Plot** window.

Part Libraries

The Part Libraries can be used to store and access a set of standard geometric entities. It is possible to create custom user-defined geometry libraries or to use built-in collections of geometric entities that are available with many add-on modules for COMSOL Multiphysics.

A dedicated part library for the Structural Mechanics Module is included. The library consists of two main parts:

- Bolts, nuts, and washers. These are parts intended for quick modeling of fasteners. The bolts geometries are prepared for use with the Modeling Pretensioned Bolts functionality. Some of the bolt geometries are also augmented with an extra domain intended for use as a "drill" when building the geometry.

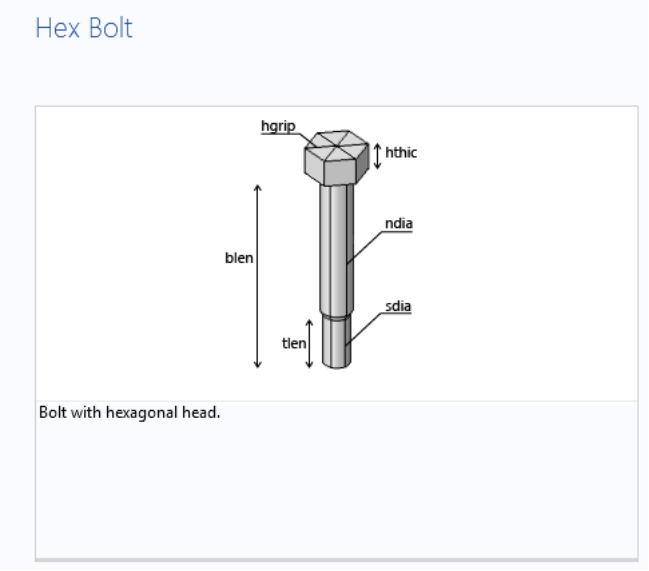


Figure 2-57: Example of a bolt part.

- Beam cross sections. These are 2D geometries, primarily intended for use in The Beam Cross Section Interface. The library includes geometries for all standard beam

cross sections according to European and US standards, as well as generic beam geometries.

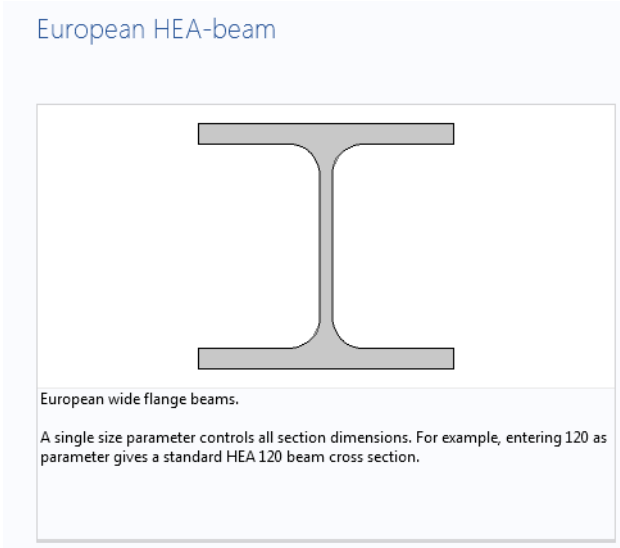


Figure 2-58: Example of a standard beam cross section.

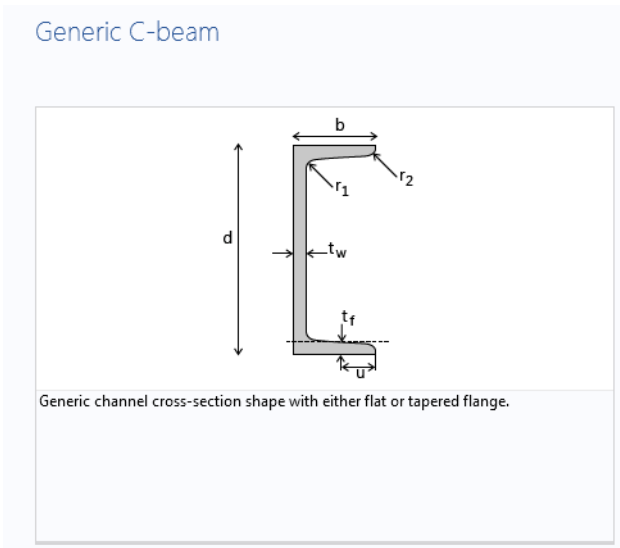


Figure 2-59: Example of a generic beam cross section.

All entities in the built-in the Part Libraries are fully parameterized, making them easy to use as parts in large-scale industrial models.

	Part Libraries in the *COMSOL Multiphysics Reference Manual.*

Estimation of Material Parameters

Hyperelastic components like seals, insulators, soft sensors, and actuators frequently exhibit a nonlinear stress–strain response. Viscoelastic models for rubber and soft biological tissues display strain-rate sensitivity and hysteresis during cyclic loading. Likewise, metals and alloys demonstrate ratcheting and nonlinear responses under cyclic loading. To ensure precise predictions of these components' behavior, it is imperative to calibrate nonlinear material models using an extensive dataset of experimental results.

With the Nonlinear Structural Materials Module, you can calibrate the model parameters of any built-in and user-defined material model to experimental data using nonlinear least-squares **Parameter Estimation** techniques. The **Parameter Estimation** functionality is available in COMSOL Multiphysics in the context menu of a **Component** or under **Optimization** in the **Physics** toolbar. Each **Global Least-Squares Objective** subnode adds an objective function Q to the model, which is of the form

$$Q = \frac{1}{2}\sum_{i=1}^{N}\frac{1}{s_i^2}(P(p_i, \mathbf{q}) - \hat{P}_i)^2 \tag{2-34}$$

Herein, p_i represents an experimental parameter (for example, applied stretch or time), $\hat{P}_i$ is the corresponding experimental value (for example, measured stress), and $P(p_i, \mathbf{q})$ is the corresponding model prediction. The quantity s_i is a scale that weighs the terms in the objective function and ensures that Q is dimensionless.

The model expression P normally depends implicitly on the model parameters $\mathbf{q}$ through the solution of the *forward* problem. To solve the *inverse* problem, the forward model and the objectives need to be combined with a study containing a **Parameter Estimation** study step. When multiple objectives are selected in the study step, the total objective function that is minimized is the sum of all the objective functions selected. During the parameter estimation study, the forward problem will be solved multiple times for different values of the model parameters, so it is good practice to first set up and test the forward model before running a parameter estimation study.

Three optimization solvers are available with the Nonlinear Structural Materials Module in the **Parameter Estimation** study step: the derivative-free **BOBYQA** solver, and the gradient-based **IPOPT** and **Levenberg–Marquardt** solvers. To avoid finding unphysical

regions in the parameter space, it is possible to specify bounds on the parameters. For most least-squares problems, the default **Levenberg–Marquardt** algorithm with a finite difference approximation of the Jacobian is a robust and efficient choice of optimization solver.

By default, the **Levenberg–Marquardt** solver is set to terminate if either the increment of the (scaled) parameters or the maximum angle between the error vector and the Jacobian is smaller than a given optimality tolerance (default to `1e-3`). In the settings of the **Optimization Solver**, it is possible to include an additional termination criterion based on the relative change of the objective function by selecting the **Terminate also for defect reduction** checkbox, which can be useful if the solver reaches a relatively flat local minimum in the parameter space. The default termination criteria are normally more robust, however.

To monitor the progress of the optimization visually, is it possible to set up a plot while solving that compares the current model prediction with the experimental data.

For further information about the **Parameter Estimation** functionality and the optimization solvers, see

- Parameter Estimation and Global Least-Squares Objective in the *Optimization Module User's Guide*.
- The Parameter Estimation Study in the *Optimization Module User's Guide*.
- The Levenberg–Marquardt Solver in the *Optimization Module User's Guide*.

For examples showing how to set up a parameter estimation study, see

- *Parameter Estimation of Hyperelastic Materials*: Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/parameter_estimation_hyperelasticity**.
- *Parameter Estimation of Elastoplastic Materials*: Application Library path **Nonlinear_Structural_Materials_Module/Plasticity/parameter_estimation_plasticity**.
- *Parameter Estimation of Viscoplastic Polymers*: Application Library path **Nonlinear_Structural_Materials_Module/Viscoplasticity/parameter_estimation_polymer_viscoplasticity**.

3

Structural Mechanics Theory

This chapter contains the theory behind the implementation of the structural mechanics functionality in COMSOL Multiphysics. To a large extent, this theory is general. Parts that are specific to certain physics interfaces, such Beam or Shell are kept in separate sections.

In this chapter:

- Solid Mechanics Theory
- Frames and Coordinate Systems
- Analysis of Deformation
- Stresses
- Equation of Motion
- Material Models
- Formulation of the Equilibrium Equations
- Study Types
- Damping
- Loads and Constraints
- Contact Analysis Theory

- Theory for Spot Welds
- Theory for Fasteners
- Bolt Modeling Theory
- Stress Linearization
- J-Integral Theory
- Embedded Elements
- Thin Layers
- Enclosed Fluids
- Reduced Integration and Hourglass Stabilization
- Average Rotation
- Warpage
- Energy Quantities
- Theory for the Shell and Plate Interfaces
- Theory for the Layered Shell Interface
- Theory for the Membrane Interface
- Theory for the Beam Interface
- Theory for the Beam Cross Section Interface
- Theory for the Truss Interface
- Theory for the Wire Interface
- Theory for the Pipe Mechanics Interface
- Theory for the Phase Field in Solids Interface
- Theory for the Transport in Solids Interface
- References

Solid Mechanics Theory

Introduction

In the following, the theory for the physics interfaces in the Solid Mechanics interface is described, as well as theory that is common for all structural mechanics interfaces.

TENSOR NOTATION

Some of the theory is developed using tensor notation. In most cases, explicit index notation is avoided. This means that the order of a tensor usually must be understood from the context. As an example, Hooke's law for linear elasticity is usually written like

$$\sigma = \mathsf{C}:\varepsilon$$

Here, the stress tensor σ and the strain tensor ε are second-order tensors, while the constitutive tensor C is a fourth-order tensor. The ':' symbol means a contraction over two indices. In a notation where the indices are shown, the same equation would read

$$\sigma_{ij} = \mathsf{C}_{ijkl}\varepsilon_{kl}$$

where the Einstein summation convention has been used as a shorthand for

$$\sigma_{ij} = \sum_{k=1}^{3}\sum_{l=1}^{3} \mathsf{C}_{ijkl}\varepsilon_{kl}$$

In a few cases, nonorthonormal coordinate systems must be considered. It is then necessary to keep track of the covariance and contravariance properties of tensors. In such a case, Hooke's law is written

$$\sigma^{ij} = \mathsf{C}^{ijkl}\varepsilon_{kl}$$

The stress and constitutive tensors have contravariant components, while the strain tensor has covariant components.

Frames and Coordinate Systems

In this section:

- Material and Spatial Coordinates
- Coordinate Systems

Material and Spatial Coordinates

The Solid Mechanics interface, through its equations, describes the motion and deformation of solid objects in a 1-, 2-, or 3-dimensional space. In COMSOL Multiphysics terminology, this physical space is known as the *spatial frame*, and positions in the physical space are identified by lowercase spatial coordinate variables x, y, and z (or r, φ, and z in axisymmetric components).

Continuum mechanics theory also makes use of a second set of coordinates, known as *material* (or *reference*) coordinates. These are normally denoted by uppercase variables X, Y, and Z (or R, Φ, and Z) and are used to label material particles. Any material particle is uniquely identified by its position in some given initial or reference configuration. As long as the solid stays in this configuration, material and spatial coordinates of every particle coincide and displacements are zero by definition.

More information can be found in About Frames chapter in the *COMSOL Multiphysics Reference Manual*.

When the solid object deforms due to external or internal forces and constraints, each material particle keeps its material coordinates $\mathbf{X}$ (bold font is used to denote coordinate vectors), while its spatial coordinates change with time and applied forces such that it follows a path

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \tag{3-1}$$

in space. Because the material coordinates are constant, the current spatial position is uniquely determined by the displacement vector $\mathbf{u}$, pointing from the reference position to the current position. The global Cartesian components of this displacement vector in the spatial frame, by default called u, v, and w, are the primary dependent variables in the Solid Mechanics interface.

By default, the Solid Mechanics interface uses the calculated displacement and Equation 3-1 to define the difference between spatial coordinates $\mathbf{x}$ and material

coordinates **X**. This means the material coordinates relate to the original geometry, while the spatial coordinates are solution dependent.

Material coordinate variables X, Y, and Z must be used in coordinate-dependent expressions that refer to positions in the original geometry, for example, for material properties that are supposed to follow the material during deformation. On the other hand, quantities that have a coordinate dependence in physical space — for example, a spatially varying electromagnetic field acting as a force on the solid — must be described using spatial coordinate variables x, y, and z.

In a geometrically linear analysis, no difference is made between the two coordinate systems. For this case, the material and spatial coordinates coincide. This may seem inconsistent with equation Equation 3-1 but ensures linearity for problems that are expected to be linear. It is then, for example, equivalent to choose a coordinate system related to the material frame or one related to the spatial frame. In a geometrically nonlinear analysis, however, any use of a spatial coordinate in an expression will introduce a nonlinear contribution because it will be deformation dependent.

Coordinate Systems

Force vectors, stress and strain tensors, as well as various material tensors are represented by their components in a specified coordinate system. By default, material properties use the canonical system in the material frame. This is the system whose basis vectors coincide with the X, Y, and Z axes. When the solid deforms, these vectors rotate with the material.

Loads and constraints, on the other hand, are applied in spatial directions, by default in the canonical spatial coordinate system. This system has basis vectors in the x, y, and z directions, which are forever fixed in space. Both the material and spatial default coordinate system are referred to as the *global coordinate system* in the physics interface.

Vector and tensor quantities defined in the global coordinate system on either frame use the frame's coordinate variable names as indices in the tensor component variable names.

For example, `solid.SXY` is the material frame *XY*-plane shear stress, also known as a second Piola–Kirchhoff stress, while `solid.sxy` is the corresponding spatial frame stress, or Cauchy stress. There are also a few mixed tensors, most notably the deformation gradient `solid.FdxY`, which has one spatial and one material index because it is used in converting quantities between the material and spatial frames.

It is possible to define any number of user coordinate systems on the material and spatial frames. Most types of coordinate systems are specified only as a rotation of the basis with respect to the canonical basis in the underlying frame. This means that they can be used both in contexts requiring a material system and in contexts requiring a spatial one. A coordinate system defined on the spatial frame will in general introduce nonlinearities in the problem, since its directions are deformation dependent in case of a geometrically nonlinear analysis.

The coordinate system can be selected separately for each added material model, load, and constraint. This is convenient when for example, an anisotropic material with different orientation in different domains is required. The currently selected coordinate system is called the *local coordinate system*.

Coordinate systems used for describing a material must be defined on the material frame.

Boundary System

Many quantities for shells, membranes, and thin layers can best be interpreted in a local coordinate system aligned to the surface. Material data, initial stresses and stress results are always represented in this local coordinate system. The local system for stress output coincides with the orientations defined for the material input.

When a Boundary System is selected, the orientation of the local system is fully defined by the boundary system.

When using a boundary system, it also possible to control the orientation of the normal by selecting the **Reverse normal direction** checkbox.

Analysis of Deformation

The analysis of deformation aims at deriving descriptions of the local deformation in a material suitable for use in a constitutive relation. Often, but not always, this amounts to deriving a *strain tensor*.

This section starts by a general description of finite deformation in solids. At the end, the specialization to engineering strains used in geometrically linear analysis, is made.

In this section:

- Lagrangian Formulation
- Deformation Measures
- Invariants of Strain
- Inelastic Strain Contributions
- Deformed Geometry
- Axial Symmetry
- Plane Strain
- Generalized Plane Strain

Lagrangian Formulation

The formulation used for structural analysis in COMSOL Multiphysics for both small and finite deformations is a *total Lagrangian* formulation. This means that the computed stress and deformation state is always referred to the material configuration rather than to current position in space.

Likewise, material properties are always given for material particles and with tensor components referring to a coordinate system based on the material frame. This has the obvious advantage that spatially varying material properties can be evaluated just once for the initial material configuration, and they do not change as the solid deforms and rotates.

Consider a certain physical particle, initially located at the coordinate $\mathbf{X}$. During deformation, this particle follows a path

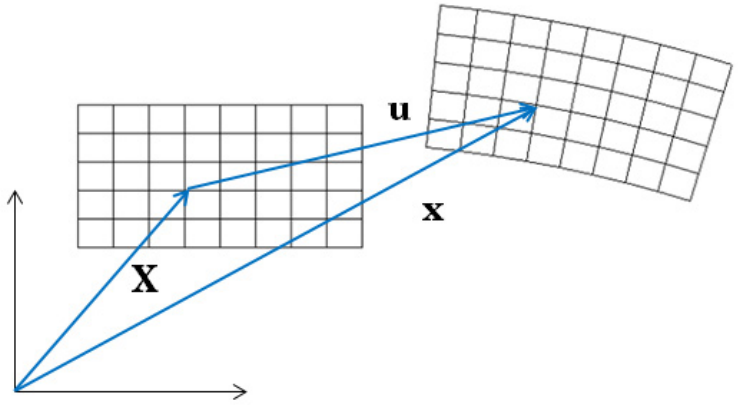
$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$$

Here, $\mathbf{x}$ is the *spatial coordinate* and $\mathbf{X}$ is the *material coordinate*.

For simplicity, assume that undeformed and deformed positions are measured in the same coordinate system. Using the displacement $\mathbf{u}$ it is then possible to write

$$\mathbf{x} = \mathbf{X} + \mathbf{u}(\mathbf{X}, t)$$

The displacement is considered as a function of the material coordinates (X, Y, Z), but it is not explicitly a function of the spatial coordinates (x, y, z). It is thus only possible to compute derivatives with respect to the material coordinates.



In the following, the gradient operator is assumed to be a gradient with respect to the material coordinates, unless something else is explicitly stated.

$$\nabla = \nabla_X = \left[\frac{\partial}{\partial X} \frac{\partial}{\partial Y} \frac{\partial}{\partial Z}\right]$$

The gradient of the displacement, which occurs frequently in the following theory, is always computed with respect to material coordinates. In 3D:

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & \frac{\partial w}{\partial Z} \end{bmatrix}$$

The *deformation gradient* tensor, $\boldsymbol{F}$, shows how an infinitesimal line element, $d\mathbf{X}$, is mapped to the corresponding deformed line element $d\mathbf{x}$ by

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} d\mathbf{X} = F d\mathbf{X}$$

The deformation gradient $\boldsymbol{F}$ contains the complete information about the local straining and rotation of the material. It is a two-point tensor (or a double vector), which transforms as a vector with respect to each of its indices. It involves both the reference and present configurations.

In terms of the displacement gradient, $\boldsymbol{F}$ can be written as

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \mathbf{u} + I$$

The deformation of the material (stretching) will in general cause changes in the material density. The ratio between current and initial volume (or mass density) is given by

$$\frac{\mathrm{d}V}{\mathrm{d}V_0} = \frac{\rho_0}{\rho} = \det(F) = J$$

Here, ρ_0 is the initial density and ρ is the current density after deformation. The determinant of the deformation gradient tensor $\boldsymbol{F}$ is related to volumetric changes with respect to the initial state. A pure rigid body displacement implies $J = 1$. Also, an incompressible material is represented by $J = 1$. These are called *isochoric* processes.

The determinant of the deformation gradient tensor is always positive (since a negative mass density is unphysical). The relation $\rho = \rho_0/J$ implies that for $J < 1$ there is compression, and for $J > 1$ there is expansion. Since $J > 0$, the deformation gradient $\boldsymbol{F}$ is invertible.

In the material formulations used within the structural mechanics interfaces, the mass density should in general be constant because the equations are formulated for fixed material particles. Thus, do not use temperature-dependent material data for the mass density. The changes in volume caused by temperature changes are incorporated using the coefficient of thermal expansion when adding Thermal Expansion (for Materials) to the material model.

The variable `solid.rho` represents a "reference" or "initial" density ρ_0, and not the "current" density ρ. If you are interested in finding the density of the deformed material (the density in the spatial frame), you can compute it using the expression `solid.rho/solid.J`.

Deformation Measures

Since the deformation tensor $\boldsymbol{F}$ is a two-point tensor, it combines both spatial and material frames. It is not symmetric. Applying a singular value decomposition on the deformation gradient tensor gives an insight into how much stretch and rotation a unit volume of material has been subjected to. The *right polar decomposition* is defined as

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$$

where $\boldsymbol{U}$ is the *right stretch tensor* given in the material frame, and $\boldsymbol{R}$ is a proper orthogonal tensor such that $\det(\boldsymbol{R}) = 1$ and $\boldsymbol{R}^{-1} = \boldsymbol{R}^{T}$. The rotation tensor $\boldsymbol{R}$ describes the rigid rotation, and all information about the deformation of the material is contained in the symmetric tensor $\boldsymbol{U}$.

- The internal variables for the deformation gradient tensor with respect to global material coordinates are named `solid.FdxX`, `solid.FdxY`, and so on.
- The internal variables for the deformation gradient tensor with respect to local material coordinates are named `solid.Fdx1`, `solid.Fdx2`, and so on.
- The rotation tensor components are named `solid.RotxX`, `solid.RotxY`, and so on.
- The right stretch tensor components are named `solid.UstchXX`, `solid.UstchXY`, and so on.

An uppercase index refers to the material frame, and a lowercase index refers to the spatial frame.

The stretch tensor contains physically important information about the deformation state. The eigenvalues of the $\boldsymbol{U}$ tensor are the principal stretches, λ_1, λ_2, and λ_3. The stretch of a line element with initial length L_0 and current length L is

$$\lambda = \frac{L}{L_0} = 1 + \varepsilon_{\text{eng}}$$

where ε_{eng} is the engineering strain. The three principal stretches act along three orthogonal directions. In the coordinate system defined by these principal directions, the $\boldsymbol{U}$ tensor will be diagonal:

$$U = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

The *right Cauchy–Green* deformation tensor C defined by

$$C = F^T F = U^2$$

It is a symmetric and positive definite tensor, which accounts for the strain but not for the rotation. The eigenvalues of the C tensor are the squared principal stretches, thus providing a more efficient way to compute the principal stretches than by using the stretch tensor U directly.

The *Green–Lagrange* strain tensor is a symmetric tensor defined as

$$\varepsilon = \frac{1}{2}(C - I) = \frac{1}{2}(F^T F - I)$$

Since C is independent of rigid body rotations, this applies also to the Green–Lagrange strain tensor.

Using the displacement components and Cartesian coordinates, the Green–Lagrange strain tensor can be written on component form as

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \cdot \frac{\partial u_k}{\partial X_j}\right). \tag{3-2}$$

The rotation independence of the Green–Lagrange strain tensor, together with the fact that it for small-strain approaches the engineering strain tensor explains why it is a common choice in constitutive models for small-strain finite rotation. As an opposite, a pure rigid rotation causes strains when engineering strains are used.

The Green–Lagrange is a natural strain representation in a Lagrangian description. Since it is a tensor in the material frame, its values should be interpreted in along the undeformed axis orientations.

- The internal variables for the Green–Lagrange strains are named `solid.eX`, `solid.eXY`, and so on.
- The internal variables for the Green–Lagrange strain tensor in local coordinates are named `solid.el11`, `solid.el12`, and so on.
- In a geometrically linear analysis, the strain variables `solid.eX`,`solid.el11`, and so on, will instead represent engineering strain.
- The right Cauchy–Green deformation tensor in local coordinate system are named `solid.Cl11`, `solid.Cl12`, and so on.

Some textbooks prefer to use the *left Cauchy–Green* deformation tensor $B = FF^T$, which is also symmetric and positive definite but it is defined in the spatial frame.

ENGINEERING STRAIN

Under the assumption of small displacements and rotations, the normal strain components and the shear strain components are related to the deformation as follows:

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_{xy} &= \frac{\gamma_{xy}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \varepsilon_y &= \frac{\partial v}{\partial y} & \varepsilon_{yz} &= \frac{\gamma_{yz}}{2} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \varepsilon_z &= \frac{\partial w}{\partial z} & \varepsilon_{xz} &= \frac{\gamma_{xz}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \end{aligned} \tag{3-3}$$

In COMSOL Multiphysics, the tensor form of strain representation (ε_{xy}, ε_{yz}, ε_{xz}) is used.

In the documentation, the symbol ε is used to denote the strain tensor in general. In a geometrically nonlinear analysis, strains should be interpreted as a Green–Lagrange strains or Hencky strains. In a geometrically linear analysis, the engineering strain is used.

The symmetric strain tensor ε consists of both normal and shear strain components:

$$\varepsilon = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix}$$

The strain–displacement relationships for the axial symmetry case for small displacements are

$$\varepsilon_r = \frac{\partial u}{\partial r}, \qquad \varepsilon_\varphi = \frac{u}{r}, \qquad \varepsilon_z = \frac{\partial w}{\partial z}, \text{ and} \qquad \gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}$$

A special problem occurs at the axis of rotation, where both u and r are zero. To avoid dividing by zero, the circumferential strain is for very small values of r redefined to

$$\varepsilon_\varphi = \frac{\partial u}{\partial r}$$

The alternative expression is obtained by applying L'Hôpital's rule.

A general description of the axially symmetric case is given in Axial Symmetry.

LOGARITHMIC STRAIN

The logarithmic strain, also called *true strain*, or *Hencky strain*, is a popular strain measure for large strain, in particular when representing data from tensile tests. For a uniaxial case, it is defined on the incremental form

$$d\varepsilon = \frac{dL}{L}$$

where L is the current length of the specimen. If this relation is integrated, the total strain can be written as

$$\varepsilon = \log\left(\frac{L}{L_0}\right) = \log\lambda$$

Here L_0 is the initial length and λ is the stretch.

In order to generalize the logarithmic strain to a strain tensor, it is necessary to first compute the three principal stretches and their orientation. Then, a logarithmic strain tensor in the local principal stretch system is defined as

$$\varepsilon_{lp} = \begin{bmatrix} \log\lambda_1 & 0 & 0 \\ 0 & \log\lambda_2 & 0 \\ 0 & 0 & \log\lambda_3 \end{bmatrix}$$

This tensor is then transformed to the global and local coordinate system to give the logarithmic strain tensor.

- The internal variables for the Hencky strain tensor with respect to global material coordinates are named `solid.eHXX`, `solid.eHXY`, and so on.
- The internal variables for the Hencky strain tensor with respect to local coordinates are named `solid.eHl11`, `solid.eHl12`, and so on.
- The internal variables for the elastic Hencky strain tensor with respect to global material coordinates are named `solid.eHeXX`, `solid.eHeXY`, and so on.
- The internal variables for the elastic Hencky strain tensor with respect to local coordinates are named `solid.eHel11`, `solid.eHel12`, and so on.
- The internal variables for the inelastic Hencky strain tensor with respect to global material coordinates are named `solid.eHieXX`, `solid.eHieXY`, and so on.
- The internal variables for the inelastic Hencky strain tensor with respect to local coordinates are named `solid.eHie11`, `solid.eHie12`, and so on.

An uppercase index refers to the global material frame, and a number refers to the local material frame.

Invariants of Strain

PRINCIPAL STRAINS

The principal strains are the eigenvalues of the strain tensor (ε), computed from the eigenvalue equation

$$(\varepsilon - \varepsilon_p I)\mathbf{v}_p = 0$$

The three principal strains are sorted so that

$$\varepsilon_{p1} \geq \varepsilon_{p2} \geq \varepsilon_{p3}$$

This sorting is true also for the 2D and 1D cases. The corresponding vectors in the principal directions, $\mathbf{v}_{pi}$, are orthonormal.

- The internal variables for the principal strains are named `solid.ep1`, `solid.ep2`, and `solid.ep3`.
- The internal variables for the components of the directions of the first principal strains are named `solid.ep1X`, `solid.ep1Y`, and `solid.ep1Z`. The direction vectors for the other two principal strains are named analogously.

The Green–Lagrange or Hencky strain tensors are used in a geometrically nonlinear analysis. The orientations of the principal directions will thus be with respect to the material directions. If you plot the principal strains as arrows, you should thus use an undeformed plot.

PRINCIPAL STRETCHES

The principal stretches are the eigenvalues of the stretch tensor U, and are also sorted by size:

$$\lambda_{p1} \geq \lambda_{p2} \geq \lambda_{p3}$$

The internal variables for the principal stretches are named `solid.stchp1`, `solid.stchp2`, and `solid.stchp3`. The elastic principal stretches are named `solid.stchelp1`, `solid.stchelp2`, and `solid.stchelp3`.

The different invariants of the strain tensor form are useful for constitutive modeling and result interpretation. The three fundamental invariants for any tensor are

$$I_1(\varepsilon) = \text{trace}(\varepsilon)$$
$$I_2(\varepsilon) = \frac{1}{2}(\text{trace}(\varepsilon)^2 - \text{trace}(\varepsilon^2))$$
$$I_3(\varepsilon) = \det(\varepsilon)$$

The invariants of the strain deviator tensor are also useful:

$$\text{dev}(\varepsilon) = \varepsilon - \frac{1}{3}\text{trace}(\varepsilon)I$$

$$J_1(\varepsilon) = \text{trace}(\text{dev}(\varepsilon)) = 0$$

$$J_2(\varepsilon) = \frac{1}{2}\text{dev}(\varepsilon):\text{dev}(\varepsilon) = \frac{1}{3}I_1(\varepsilon)^2 - I_2(\varepsilon)$$

$$J_3(\varepsilon) = \det(\text{dev}(\varepsilon)) = \frac{2}{27}I_1(\varepsilon)^3 - \frac{1}{3}I_1(\varepsilon)I_2(\varepsilon) + I_3(\varepsilon)$$

As defined above $J_2 \geq 0$. I_1 represents the relative change in volume for infinitesimal strains and J_2 represents the magnitude of shear strain.

In tensor component notation, the invariants can be written as

$$\text{trace}(\varepsilon) = \varepsilon_{kk}$$

$$\text{dev}(\varepsilon)_{ij} = \varepsilon_{ij} - \frac{\varepsilon_{kk}}{3}\delta_{ij}$$

$$I_2(\varepsilon) = \frac{1}{2}(\varepsilon_{ii}\varepsilon_{jj} - \varepsilon_{ij}\varepsilon_{ji})$$

$$J_2(\varepsilon) = \frac{1}{2}\text{dev}(\varepsilon)_{ij}\text{dev}(\varepsilon)_{ji}$$

The volumetric strain is defined as

$$\varepsilon_{\text{vol}} = \text{trace}(\varepsilon)$$

and the equivalent deviatoric strain as

$$\varepsilon_{\text{deve}} = \sqrt{\frac{2}{3}\text{dev}(\varepsilon)_{ij}\text{dev}(\varepsilon)_{ji}} = \sqrt{\frac{4}{3}J_2(\varepsilon)}$$

The internal variable for the volumetric strain is `solid.evol`. The internal variables for the components of the deviatoric strain tensor in the local coordinate system are `solid.eldev11`, and so on. The internal variable for the equivalent deviatoric strain is `solid.edeve`.

In terms of the principal strains, the strain invariants can be written as

$$I_1(\varepsilon) = \varepsilon_{p1} + \varepsilon_{p2} + \varepsilon_{p3}$$
$$I_2(\varepsilon) = \varepsilon_{p1}\varepsilon_{p2} + \varepsilon_{p2}\varepsilon_{p3} + \varepsilon_{p1}\varepsilon_{p3}$$
$$I_3(\varepsilon) = \varepsilon_{p1}\varepsilon_{p2}\varepsilon_{p3}$$

The principal strains are the roots of the *characteristic equation* (Cayley–Hamilton theorem)

$$\varepsilon_p^{\,3} - I_1\varepsilon_p^{\,2} + I_2\varepsilon_p - I_3 = 0$$

STRAIN RATE AND SPIN

The spatial velocity gradient is defined in components as

$$L_{kl} = \frac{\partial}{\partial x_l} v_k(\mathbf{x}, t)$$

where $v_k(\mathbf{x},t)$ is the spatial velocity field. It can be shown that L can be computed in terms of the deformation gradient as

$$L = \frac{dF}{dt} F^{-1}$$

where the material time derivative is used.

The velocity gradient can be decomposed into symmetric and skew-symmetric parts

$$L = L_d + L_w$$

where

$$L_d = \frac{1}{2}(L + L^T)$$

is called the *rate of strain tensor*, and

$$L_w = \frac{1}{2}(L - L^T)$$

is called the *spin tensor*. Both tensors are defined on the spatial frame.

It can be shown that the material time derivative of the Green–Lagrange strain tensor can be related to the rate of strain tensor as

$$\frac{d\varepsilon}{dt} = F^T L_d F$$

The spin tensor $L_w(\mathbf{x},t)$ accounts for an instantaneous local rigid-body rotation about an axis passing through the point x.

Components of both L_d and L_w are available as results and analysis variables under the Solid Mechanics interface.

- The components of the rate of strain tensor are named `solid.Ldx`, `solid.Ldxy`, and so on.
- The components of the spin tensor are named `solid.Lwx`, `solid.Lwxy`, and so on.

Inelastic Strain Contributions

Many of the material models in COMSOL Multiphysics will compute a stress based on an *elastic strain*. The elastic strain tensor is obtained after removing any inelastic deformation contribution from the total deformation from the displacements. There are several possible inelastic strain contributions:

- Initial strain, ε_0
- External strain, ε_{ex}
- Thermal strain, ε_{th}
- Hygroscopic strain, ε_{hs}
- Plastic strain, ε_{pl}
- Creep strain, ε_{cr}
- Viscoplastic strain, ε_{vp}
- Viscoelastic strain, ε_{ve}

ADDITIVE DECOMPOSITION

In a geometrically linear analysis, the elastic strain is computed by a straightforward subtraction of the inelastic strain:

$$\varepsilon_e = \varepsilon - \varepsilon_{inel}$$

where

$$\varepsilon_{inel} = \varepsilon_0 + \varepsilon_{ex} + \varepsilon_{th} + \varepsilon_{hs} + \varepsilon_{pl} + \varepsilon_{cr} + \varepsilon_{vp} + \varepsilon_{ve}$$

Additive decomposition of strains can also be used in a geometrically nonlinear analysis. In this case, it can be justified as long as the strains are small, or when using Hencky (logarithmic) strains. In the case of large deformations, the different strain contributions may not even be commutative.

Choose to use additive decomposition also for geometric nonlinearity in the **Geometric Nonlinearity** section in the settings for Linear Elastic Material or Nonlinear Elastic Material.

MULTIPLICATIVE DECOMPOSITION

In the finite deformation case, the inelastic strain can be removed by using a multiplicative decomposition of the deformation gradient tensor. The *elastic deformation* gradient tensor is the basis for all strain energy formulations in hyperelastic materials, and also for the elastic strain in linear and nonlinear elasticity. It is derived by removing the inelastic deformation from the total deformation gradient tensor.

The total deformation gradient tensor is defined as the result of two successive operations, an inelastic deformation followed by an elastic deformation:

$$\boldsymbol{F} = \boldsymbol{F}_{el}\boldsymbol{F}_{inel} \tag{3-4}$$

The order is important here, multiplication from the left makes the elastic deformation act on the inelastically deformed state.

Since a deformation gradient tensor describes a mapping from one frame to another, there are actually three frames involved in this operation. The $\boldsymbol{F}$ tensor is defined by the displacements as usual and describes the mapping from the material frame to the spatial frame. The $\boldsymbol{F}_{inel}$ tensor, however, describes a mapping from the material frame to an intermediate frame, and the $\boldsymbol{F}_{el}$ tensor describes a mapping from the intermediate frame to the spatial frame.

When the inelastic deformation gradient tensor is known, the elastic deformation gradient tensor is computed as

$$\boldsymbol{F}_{el} = \boldsymbol{F}\boldsymbol{F}_{inel}^{-1} \tag{3-5}$$

so the inelastic deformations are removed from the total deformation gradient tensor. The inelastic deformation tensor $\boldsymbol{F}_{\text{inel}}$ is derived from inelastic processes, such as thermal expansion or plasticity. When there are several inelastic contributions, they are applied sequentially to obtain the total inelastic deformation tensor $\boldsymbol{F}_{\text{inel}}$:

$$\boldsymbol{F}_{\text{inel}} = \boldsymbol{F}_1\boldsymbol{F}_2\boldsymbol{F}_3\boldsymbol{F}_4 \ \cdots$$

where $\boldsymbol{F}_i$ is the inelastic strain contribution from subnode i under a Linear Elastic Material, Nonlinear Elastic Material, or Hyperelastic Material.

- The order is important when deformations are finite. The contributions are applied in the same order as the subnodes appear in the model tree. If a **Thermal Expansion** node appears before a **Plasticity** node, then the physical process can be viewed as a thermal expansion followed by a plastic deformation.
- When a certain inelastic strain contribution is small, the order is not significant.
- If the inelastic strain is a pure isotropic volume change, as is often the case for thermal expansion and hygroscopic swelling, the order is not significant.

The *right Cauchy–Green* deformation tensor $\boldsymbol{C}$ is computed from the deformation gradient $\boldsymbol{F}$, as well as the *Green–Lagrange* strain tensor ε

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} \text{ and } \varepsilon = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$$

The *elastic right Cauchy–Green* deformation tensor is then computed from $\boldsymbol{F}_{\text{el}}$

$$\boldsymbol{C}_{\text{el}} = \boldsymbol{F}_{\text{el}}^T\boldsymbol{F}_{\text{el}}$$

and the *elastic Green–Lagrange* strain tensor is computed as

$$\varepsilon_{el} = \frac{1}{2}(C_{el} - I)$$

The internal variables for the elastic right Cauchy–Green deformation tensor in the local coordinate system are named `solid.Cel11`, `solid.Cel12`, and so on; and for the elastic Green–Lagrange tensor in local coordinates `solid.eel11`, `solid.eel12`, and so on.

The elastic, inelastic, and total volume ratios are related as

$$\det(F) = \det(F_{cl})\det(F_{inel}) \text{ or } J = J_{cl}J_{inel}$$

The internal variables for the elastic, inelastic, and total volume ratio are named `solid.Jel`, `solid.Ji`, and `solid.J`.

After Equation 3-5, the elastic Green–Lagrange strain depends on the inelastic deformation as

$$\varepsilon_{el} = \frac{1}{2}(C_{el} - I) = \frac{1}{2}F_{inel}^{-T}(C - C_{inel})F_{inel}^{-1} = F_{inel}^{-T}(\varepsilon - \varepsilon_{inel})F_{inel}^{-1} \qquad (3\text{-}6)$$

where the *inelastic Green–Lagrange* strain reads

$$\varepsilon_{inel} = \frac{1}{2}(C_{inel} - I) = \frac{1}{2}(F_{inel}^{T}F_{inel} - I)$$

When using multiplicative decomposition of deformation gradients, Hooke's law for a Linear Elastic Material produces a second Piola–Kirchhoff stress tensor which is linearly related to the elastic Green–Lagrange strain

$$S_{el} = \mathsf{C} : \varepsilon_{el}$$

This is however, a stress tensor defined in the intermediate configuration. Introducing a pull-back operation of this tensor, Hooke's law for the multiplicative decomposition of elastic and inelastic strains reads

$$S = J_{inel}F_{inel}^{-1}\mathsf{C} : (\varepsilon - \varepsilon_{inel})F_{inel}^{-T}$$

LOGARITHMIC DECOMPOSITION

For finite deformations, the inelastic strain can be removed by using an additive decomposition of logarithmic stretches. The *elastic Hencky* tensor is the basis for the strain energy formulation. It is computed by removing the inelastic strains from the total *Hencky* strain tensor:

$$\varepsilon_{\mathrm{Hel}} = \varepsilon_{\mathrm{H}} - \varepsilon_{\mathrm{inel}} \tag{3-7}$$

here, the inelastic logarithmic strain, $\varepsilon_{\mathrm{inel}}$, is computed as the accumulation of all possible inelastic effects

$$\varepsilon_{\mathrm{inel}} = \varepsilon_0 + \varepsilon_{\mathrm{ex}} + \varepsilon_{\mathrm{th}} + \varepsilon_{\mathrm{hs}} + \varepsilon_{\mathrm{pl}} + \varepsilon_{\mathrm{cr}} + \varepsilon_{\mathrm{vp}} + \varepsilon_{\mathrm{ve}}$$

The *right Cauchy–Green* deformation tensor C is computed from the deformation gradient F, as well as the total *Hencky* strain tensor ε_{H}

$$C = F^T F \text{ and } \varepsilon_{\mathrm{H}} = \frac{1}{2}\log(C)$$

The Padé approximation (Ref. 51) of the matrix logarithm reads

$$\varepsilon_{\mathrm{H}} = \frac{1}{2}\log(C) \approx \frac{3}{2}(C^2 - I)(C^2 + 4C + I)^{-1}$$

which can be faster than the analytical formula for moderate stretches.

When using an additive decomposition of logarithmic strains, Hooke's law for a Linear Elastic Material produces a Mandel stress tensor which is linearly related to the elastic Hencky strain

$$S_{\mathrm{el}} = \mathsf{C} : \varepsilon_{\mathrm{Hel}}$$

Deformed Geometry

The equations solved by the Solid Mechanics interface are formulated in the material frame. The Deformed Geometry functionality in COMSOL Multiphysics allows one to make the material frame differ from the geometry frame, which implies that the geometry of the structure on the material frame can differ from that originally drawn.

This is useful for analyzing the behavior of different shapes of an original object, for example as part of shape optimization.

In the *COMSOL Multiphysics Reference Manual*:

Deformed Geometry and Moving Mesh

By default, the material does not follow a change in shape. Deformation of the geometric boundaries therefore corresponds to addition or removal of material.

The deformed geometry functionality can be also used to set up an incremental deformation of a structure. This can be achieved by using the **Elastic Predeformation** node in the Solid Mechanics interface. This can significantly speed up computations is case of large deformations.

ELASTIC PREDEFORMATION

The total displacement field of the solid is represented as

$$\mathbf{u}_{\text{tot}} = \mathbf{u}_{\text{pd}} + \mathbf{u}$$

and the corresponding deformation gradient is multiplicatively decomposed as

$$\boldsymbol{F}_{\text{tot}} = \boldsymbol{F}\boldsymbol{F}_{\text{pd}}$$

One can use $\mathbf{u}_{\text{pd}}$ to define the displacement of the material frame with respect to the geometry frame, and take $\mathbf{u}$ as an update solution of the equations solved by the Solid Mechanics interface on the material frame. (The corresponding deformation gradient $\boldsymbol{F}$ connects the material and spatial frames.) Note that the virtual work on such a displacement update is done by the total stress, and to get the correct stress, one can use as $\boldsymbol{F}_{\text{pd}}$ the deformation gradient connecting the geometry and material frames, and then set the elastic deformation gradient to

$$\boldsymbol{F}_{\text{el}} = \boldsymbol{F}_{\text{tot}} \tag{3-8}$$

This is used in the stress calculation during the update step. Thus, the transformation from geometry to material frame is assumed to be a result of elastic deformation.

The process can be repeated in a parametric sweep, for example, by gradually ramping up the load, so that the displacement update on each step is small. During each step, the previous value of the total displacement is used in the material frame definition

$$\mathbf{u}_{\text{pd}} = \mathbf{u}_{\text{tot, old}}$$

Alternatively, a time-dependent study can be used. In this situation, the above formula will be applied during each time step to recompute $\mathbf{u}_{\mathrm{pd}}$ before the step, using the stored previous solution, while $\mathbf{u}$ will represent a displacement increment computed during the time step. In addition, the density is updated as ρ_0/J_{pd} where

$$J_{\mathrm{pd}} = \det(\boldsymbol{F}_{\mathrm{pd}})$$

Again, $\boldsymbol{F}_{\mathrm{pd}}$ is the deformation gradient connecting the geometry and material frames, and it will be updated before each time step or parameter sweep value.

The approach can be combined with large strain plasticity. In this case,

$$\boldsymbol{F}_{\mathrm{tot}} = \boldsymbol{F}_{\mathrm{el}}\boldsymbol{F}_{\mathrm{pl}}$$

which together with Equation 3-8 gives

$$\boldsymbol{F}_{\mathrm{el}} = \boldsymbol{F}\boldsymbol{F}_{\mathrm{pd}}\boldsymbol{F}_{\mathrm{pl}}^{-1}$$

Note that both $\boldsymbol{F}_{\mathrm{el}}$ and $\boldsymbol{F}_{\mathrm{pl}}$ represent the total deformations of their corresponding types, while $\boldsymbol{F}_{\mathrm{pd}}$ is the total (elastic and plastic) deformation gradient from the previous update.

Axial Symmetry

The 2D axisymmetric implementation in COMSOL Multiphysics by default assumes independence of the azimuthal component of the displacement. Therefore, the physical components of the radial and axial displacement, u and w, are used by default as dependent variables for the axially symmetric geometry. It is also possible to include the dependent variable v for the out-of-plane displacement, or an azimuthal mode extension in time-harmonic studies. See Circumferential Displacement and Circumferential Modes.

STRAINS

The displacement gradient with respect to the cylindrical coordinates of the undeformed geometry reads

$$\nabla \mathbf{u} = \begin{bmatrix} \dfrac{\partial u}{\partial R} & \dfrac{1}{R}\dfrac{\partial u}{\partial \Phi} - \dfrac{v}{R} & \dfrac{\partial u}{\partial Z} \\ \dfrac{\partial v}{\partial R} & \dfrac{1}{R}\dfrac{\partial v}{\partial \Phi} + \dfrac{u}{R} & \dfrac{\partial v}{\partial Z} \\ \dfrac{\partial w}{\partial R} & \dfrac{1}{R}\dfrac{\partial w}{\partial \Phi} & \dfrac{\partial w}{\partial Z} \end{bmatrix}$$

The assumption of axial symmetry neglect gradients in the azimuthal direction, so the displacement vector is considered independent on the azimuthal angle, $\mathbf{u} = \mathbf{u}(R, Z)$, and

$$\partial u/\partial \Phi = \partial v/\partial \Phi = \partial w/\partial \Phi = 0 .$$

The displacement gradient after this assumption reads

$$\nabla \mathbf{u} = \begin{bmatrix} \dfrac{\partial u}{\partial R} & -\dfrac{v}{R} & \dfrac{\partial u}{\partial Z} \\ \dfrac{\partial v}{\partial R} & \dfrac{u}{R} & \dfrac{\partial v}{\partial Z} \\ \dfrac{\partial w}{\partial R} & 0 & \dfrac{\partial w}{\partial Z} \end{bmatrix}$$

and the Green–Lagrange strain tensor is

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u} + \nabla \mathbf{u}^T \nabla \mathbf{u})$$

Assuming that there is no torsion around the axis of symmetry, so there is no out-of-plane displacement, $v = 0$, the deformation gradient can be further simplified to read

$$\nabla \mathbf{u} = \begin{bmatrix} \dfrac{\partial u}{\partial R} & 0 & \dfrac{\partial u}{\partial Z} \\ 0 & \dfrac{u}{R} & 0 \\ \dfrac{\partial w}{\partial R} & 0 & \dfrac{\partial w}{\partial Z} \end{bmatrix}$$

For geometrically linear analysis, the nonlinear terms in the Green–Lagrange strain tensor are dropped

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u}) = \begin{bmatrix} \frac{\partial u}{\partial R} & 0 & \frac{1}{2}\left(\frac{\partial u}{\partial Z} + \frac{\partial w}{\partial R}\right) \\ 0 & \frac{u}{R} & 0 \\ \frac{1}{2}\left(\frac{\partial u}{\partial Z} + \frac{\partial w}{\partial R}\right) & 0 & \frac{\partial w}{\partial Z} \end{bmatrix}$$

and the volumetric strain is computed from

$$\varepsilon_{\text{vol}} = I_1(\varepsilon) = \frac{\partial u}{\partial R} + \frac{u}{R} + \frac{\partial w}{\partial Z}$$

For the 1D axisymmetric representation, only the radial component of the displacement field is computed, and only gradients with respect to the radial direction are considered, this is, $u = u(R)$, and $\partial u/\partial \Phi = \partial u/\partial Z = 0$.

CIRCUMFERENTIAL DISPLACEMENT

When the out-of-plane displacement is considered in a 2D axisymmetric model, torsion is allowed with respect to the symmetry axis. In this case, the displacement gradient reads

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial R} & -\frac{v}{R} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial R} & \frac{u}{R} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial R} & 0 & \frac{\partial w}{\partial Z} \end{bmatrix}$$

and the Green–Lagrange strain tensor is

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u} + \nabla \mathbf{u}^T \nabla \mathbf{u})$$

For geometrically linear analysis, the nonlinear terms in the Green–Lagrange strain tensor are dropped

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u}) = \begin{bmatrix} \frac{\partial u}{\partial R} & \frac{1}{2}\left(\frac{\partial v}{\partial R} - \frac{v}{R}\right) & \frac{1}{2}\left(\frac{\partial u}{\partial Z} + \frac{\partial w}{\partial R}\right) \\ \frac{1}{2}\left(\frac{\partial v}{\partial R} - \frac{v}{R}\right) & \frac{u}{R} & \frac{1}{2}\frac{\partial v}{\partial Z} \\ \frac{1}{2}\left(\frac{\partial u}{\partial Z} + \frac{\partial w}{\partial R}\right) & \frac{1}{2}\frac{\partial v}{\partial Z} & \frac{\partial w}{\partial Z} \end{bmatrix}$$

and the volumetric strain is computed from

$$\varepsilon_{\text{vol}} = I_1(\varepsilon) = \frac{\partial u}{\partial R} + \frac{u}{R} + \frac{\partial w}{\partial Z}$$

CIRCUMFERENTIAL MODES

A standard 2D axisymmetric representation of the structure assumes the independence of the solution with respect to the azimuthal angle ϕ. The following 3D solution represents an extension of this assumption:

$$\mathbf{u}(R, \phi, Z, t) = \mathbf{u}_0(R, Z) + \mathbf{u}_1(R, Z)\exp(i\omega t - im\phi)$$

where m is a *circumferential mode number* (or *azimuthal mode number*) that can only have integer values to obey the axially symmetric nature of the corresponding 3D problem, that is, there is an azimuthal symmetry $\phi = \phi + 2\pi$. Thus,

$$\mathbf{u}(R, \phi + 2\pi, Z, t) = \mathbf{u}(R, \phi, Z, t)$$

The static prestress solution $\mathbf{u}_0$ can be obtained using a standard static analysis in 2D axially symmetric geometry; and the circumferential wave number $k_m = m/R$ can be introduced to describe the circumferential modes.

The displacement vector $\mathbf{u}_1$ can have nonzero values in all three components, which are functions of the radial and axial positions. For a given circumferential mode number m, the displacement vector $\mathbf{u}_1$ can be found using an eigenfrequency analysis in a 2D axisymmetric geometry. Hence,

$$\omega = 2\pi f$$
$$f = f(m, \mathbf{u}_0)$$

and the perturbation solution becomes

$$\mathbf{u}_1 = \mathbf{u}_1(R, Z)\exp[2\pi i f(m, \mathbf{u}_0)t - im\phi]$$

The solution $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_1$ represents eigenmodes in the corresponding 3D structure, which can be computed assuming certain constraints on the axis and possible static prestress and independent of the position along the axis.

Plane Strain

For two-dimensional problems, there are tree possible approximations: *plane strain*, Generalized Plane Strain, and Plane Stress. The selection is made in the settings for the Solid Mechanics node.

For plane strain, the deformation occurs in the xy-plane and it is fully characterized by the in-plane displacement components $u(X,Y)$ and $v(X,Y)$. There is no out-of-plane displacement, $w = 0$, and there are only in-plane strains. In this case, the displacement gradient reads

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} \end{bmatrix}$$

and the strain tensor $\varepsilon_{ij} = \varepsilon_{ij}(X, Y)$ only has in-plane components:

$$\varepsilon = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_y \end{bmatrix}$$

For the 1D plane strain representation, only the x-component of the displacement field is computed, and only gradients with respect to the x direction are considered, this is, $u = u(X)$, $\partial u/\partial Y = \partial u/\partial Z = 0$, and $\varepsilon_{xy} = \varepsilon_y = 0$.

Generalized Plane Strain

One possible extension of the Plane Strain formulation is to assume that the displacement field depends on the out-of-plane coordinate Z, but in-plane strains are independent of it.

The above assumption have the following 3D solution for the displacement field:

$$u = u_0(X, Y) - \frac{a}{2}Z^2$$
$$v = v_0(X, Y) - \frac{b}{2}Z^2$$
$$w = (aX + bY + c)Z$$

here, $u_0(X,Y)$ and $v_0(X,Y)$ are the in-plane displacement components; and a, b, and c are constants independent of the X, Y, and Z coordinates. The gradient of the displacement field then reads:

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_0}{\partial X} & \frac{\partial u_0}{\partial Y} & -aZ \\ \frac{\partial v_0}{\partial X} & \frac{\partial v_0}{\partial Y} & -bZ \\ aZ & bZ & aX + bY + c \end{bmatrix}$$

At the cross section $Z = 0$, the in-plane deformation is fully characterized by the in-plane displacement components $u_0(X,Y)$ and $v_0(X,Y)$. The displacement gradient then simplifies to

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_0}{\partial X} & \frac{\partial u_0}{\partial Y} & 0 \\ \frac{\partial v_0}{\partial X} & \frac{\partial v_0}{\partial Y} & 0 \\ 0 & 0 & aX + bY + c \end{bmatrix}$$

The out-of-plane shear strains are zero, $\varepsilon_{xz} = \varepsilon_{yz} = 0$, and under the assumption of small displacements and rotations the normal strain reads

$$\varepsilon_z = ax + by + c \tag{3-9}$$

The above conditions differ from the Plane Strain formulation only by the fact that the out-of-plane strain component ε_z can vary linearly throughout the cross section.

The generalized plane strain approximation is good when the structure is free to expand in the out-of-plane direction, and the possible bending curvature is small with respect to the extents of the structure in the xy-plane. In the case where there is no out-of-plane bending, the out-of-plane strain component simplifies to $\varepsilon_z = c$.

In COMSOL Multiphysics, the coefficients a, b, and c in Equation 3-9 are modeled as extra degrees of freedom that are constant throughout the model (global variables).

For the 1D representation, only the x-component of the displacement field is considered, and only gradients with respect to this direction are computed, this is, $u = u(X)$, $\partial u/\partial Y = \partial u/\partial Z = 0$, and $\varepsilon_{xy} = \varepsilon_y = 0$. It is possible to apply the generalized plane strain assumption to either the xy-plane, the xz-plane, or to both planes, in which case the strain components are augmented to $\varepsilon_y = b$ and $\varepsilon_z = c$.

For the 1D axisymmetric representation, only the radial component of the displacement field is considered, and only gradients with respect to this direction are computed, this is, $u = u(R)$, and $\partial u/\partial \Phi = \partial u/\partial Z = 0$. It is possible to apply the generalized plane strain assumption to the $r\phi$-plane, so the strain is augmented to $\varepsilon_z = c$.

GEOMETRIC NONLINEARITY

In case of geometric nonlinearity, the strains are represented by the Green–Lagrange strain tensor:

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u} + \nabla \mathbf{u}^T \nabla \mathbf{u})$$

Consider the following displacement field expressed in terms of the material coordinates:

$$\mathbf{u} = \begin{bmatrix} u(X,Y) - \frac{a}{2}Z^2 \\ v(X,Y) - \frac{b}{2}Z^2 \\ (a(X+u) + b(Y+v) + c)Z \end{bmatrix}$$

Coefficients a, b, and c are assumed to be small. Then, using the above displacement field in the displacement gradient evaluated in the plane $Z = 0$ simplifies to

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & 0 \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & 0 \\ 0 & 0 & a(X+u) + b(Y+v) + c \end{bmatrix}$$

We obtain the strain components by dropping quadratic and higher-order terms in the coefficients a, b, and c:

$$\varepsilon_{XX} = u_X + \frac{1}{2}(u_X^2 + v_X^2)$$
$$\varepsilon_{YY} = v_Y + \frac{1}{2}(u_Y^2 + v_Y^2)$$
$$\varepsilon_{XY} = \frac{1}{2}(u_Y + v_X + u_X u_Y + v_X v_Y)$$
$$\varepsilon_{XZ} = \varepsilon_{YZ} = 0$$
$$\varepsilon_{ZZ} = a(X + u) + b(Y + v) + c$$

Thus, in the leading order approximation, the strains become independent of the out-of-plane coordinate Z.

OUT-OF-PLANE WAVES

When a 2D plane strain model represents a cross-section of the structure that has a significant uniform extension in the out-of-plane Z direction, the following 3D solution can be sought in form of the amplitude expansion:

$$\mathbf{u}(X, Y, Z, t) = \mathbf{u}_0(X, Y) + \mathbf{u}_1(X, Y)\exp(i\omega t - ik_Z Z)$$

The first term, $\mathbf{u}_0$, represents a static in-plane prestress deformation:

$$\mathbf{u}_0 = \begin{bmatrix} u_0(X, Y) \\ v_0(X, Y) \\ 0 \end{bmatrix}$$

This can be obtained by a standard static analysis using a 2D geometry for the cross section with the corresponding boundary conditions.

The second part of the solution, $\mathbf{u}_1$, presents a time-harmonic linear perturbation with an amplitude that can be a function of the in-plane coordinates X and Y. Such a perturbation can be seen as an out-of-plane wave, with a small amplitude that propagates in the Z direction, a wavelength L, and phase velocity c:

$$k_Z = 2\pi/L$$
$$c = \omega/k_Z$$
$$\mathbf{u}_1 = \mathbf{u}_1(X, Y)\exp\left[\frac{2\pi i}{L}(ct - Z)\right]$$

Note that in contrast to the prestress solution $\mathbf{u}_0$, the perturbation amplitude $\mathbf{u}_1$ can have nonzero values in all the displacement components:

$$\mathbf{u}_1 = \begin{bmatrix} u_1(X, Y) \\ v_1(X, Y) \\ w_1(X, Y) \end{bmatrix}$$

There are two alternative approaches. The wavelength L, and thus the wave number k_Z, can be considered as a parameter. Then, ω can be computed by an eigenfrequency analysis for the 2D cross section with all three displacement components taken as dependent variables. As a result, one obtains

$$\omega = 2\pi f$$
$$f = f(L, \mathbf{u}_0)$$
$$c(\lambda, \mathbf{u}_0) = Lf(L, \mathbf{u}_0)$$

Thus, the wave speed for the out-of-plane wave is computed as a function of the wavelength L and possible prestress $\mathbf{u}_0$ in the material. The dependence of the wave speed on the wavelength is often called dispersion.

Alternatively, the frequency f (and thus ω) can be taken as a parameter. Then, the solution can be computed via eigenvalue analysis with respect to the wave number k_Z using the 2D cross section geometry. Hence,

$$k_Z = 2\pi/L$$
$$L = L(f, \mathbf{u}_0)$$
$$c(f, \mathbf{u}_0) = fL(f, \mathbf{u}_0)$$

which determine the wavelength L and phase velocity c for the wave that propagates out-of-plane for a given frequency f under given in-plane prestress deformation $\mathbf{u}_0$. Such interpretation of the perturbation solution is sometimes called a signaling problem.

Stresses

In this section:

- Defining Stress
- Invariants of the Stress Tensor
- Plane Stress
- Initial Stress and Strain
- Axial Symmetry and Stresses

Defining Stress

This section summarizes the definition of different stress measures, stress invariants, and other important definitions.

Three different stress measures are used in COMSOL Multiphysics:

- Cauchy stress σ defined as force/deformed area in fixed spatial directions not following the body. This is a symmetric tensor.
- First Piola–Kirchhoff stress P. The forces in the spatial directions are related to the area in the original (material) frame. This is an unsymmetric two-point tensor.
- Second Piola–Kirchhoff stress S. Both force and area are represented in the material configuration. For small strains the values are the same as Cauchy stress tensor but the directions are rotating with the body. This is a symmetric tensor.

The stresses relate to each other as

$$S = F^{-1}P$$
$$\sigma = J^{-1}PF^{T} = J^{-1}FSF^{T}$$

In a geometrically linear analysis, the distinction between the stress measures disappear, and they all converge to the same values.

In the documentation, the symbol σ is used to denote not only Cauchy stress, but stress in general. The symbols P and S are used whenever it is necessary to make a distinction. In geometrically nonlinear analysis, the stress should in general be interpreted as second Piola–Kirchhoff stress.

- The internal variables for the Cauchy stresses are named `solid.sx`, `solid.sxy`, and so on.
- The internal variables for the first Piola–Kirchhoff stresses are named `solid.PxX`, `solid.PxY`, and so on.
- The internal variables for the second Piola–Kirchhoff stresses are named `solid.SX`, `solid.SXY`, and so on.

SIGN CONVENTIONS

A positive normal stress in COMSOL Multiphysics acts in tension. This is the most widely used definition in general physics and engineering.

Within the field of geomechanics it is however common to let compressive stresses be positive, since compression is almost always dominant in that field of science. When working with the material models intended for soils and rocks, you must be aware of that the "positive in tension" convention is used also there in order to maintain consistency within the software.

Specifically, the ordering of Principal Stresses is such that $\sigma_{p1} \geq \sigma_{p2} \geq \sigma_{p3}$ (including signs). In geotechnical applications the dominant principal stresses will usually be compressive, so the third principal stress will the be the one which you may consider as "largest".

The convention used in Ref. 54 refers to the hydrostatic pressure (trace of the stress Cauchy tensor) with a positive sign. The use of the first invariant of Cauchy stress tensor $I_1(\sigma)$ is preferred in this document where there is a risk of confusion. The pressure in COMSOL Multiphysics is always defined as positive under compression, or equivalently, it has the opposite sign of the Cauchy stress tensor's trace.

Invariants of the Stress Tensor

The different invariants of the stress tensor form an important basis for constitutive models and also for interpretation of stress results. The three fundamental invariants for any tensor are

$$\begin{aligned} I_1(\sigma) &= \text{trace}(\sigma) \\ I_2(\sigma) &= \frac{1}{2}(\text{trace}(\sigma)^2 - \text{trace}(\sigma^2)) \\ I_3(\sigma) &= \det(\sigma) \end{aligned} \tag{3-10}$$

In many cases, the invariants of the deviatoric stress tensor $\mathrm{dev}(\sigma) = \sigma - \frac{1}{3}\mathrm{trace}(\sigma)I$ are also useful.

$$J_2(\sigma) = \frac{1}{2}\mathrm{dev}(\sigma):\mathrm{dev}(\sigma) = \frac{1}{3}I_1(\sigma)^2 - I_2(\sigma)$$
$$J_3(\sigma) = \det(\mathrm{dev}(\sigma)) = \frac{2}{27}I_1(\sigma)^3 - \frac{1}{3}I_1(\sigma)I_2(\sigma) + I_3(\sigma) \quad (3\text{-}11)$$

As defined above $J_2 \geq 0$. In many material models, the most relevant invariants are I_1, J_2, and J_3. I_1 represents the effect of mean stress, J_2 represents the magnitude of shear stress, and J_3 contains information about the direction of the shear stress.

In tensor component notation, the invariants can be written as

$$\mathrm{trace}(\sigma) = \sigma_{kk}$$
$$\mathrm{dev}(\sigma)_{ij} = \sigma_{ij} - \frac{\sigma_{kk}}{3}\delta_{ij}$$
$$I_2(\sigma) = \frac{1}{2}(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji})$$
$$J_2(\sigma) = \frac{1}{2}\mathrm{dev}(\sigma)_{ij}\mathrm{dev}(\sigma)_{ji}$$

The pressure is defined as

$$p = \frac{-\mathrm{trace}(\sigma)}{3} = \frac{-I_1(\sigma)}{3}$$

and is thus positive in compression.

- The internal variables for the invariants I_1, I_2, and I_3 are named `solid.I1s`, `solid.I2s`, and `solid.I3s`, respectively.
- The internal variables for the invariants J_2 and J_3 are named `solid.II2s` and `solid.II3s`, respectively.
- The internal variable for the pressure is `solid.pm`.

PRINCIPAL STRESSES

The principal stresses are the eigenvalues of the stress tensor, computed from the eigenvalue equation.

$$(\sigma - \sigma_p I)\mathbf{v}_p = 0$$

The three principal stresses are ordered so that

$$\sigma_{p1} \geq \sigma_{p2} \geq \sigma_{p3}$$

This ordering is true also for the 2D cases. The corresponding principal directions $\mathbf{v}_{pi}$ are orthonormal.

- The internal variables for the principal stresses are named `solid.sp1`, `solid.sp2`, and `solid.sp3`.
- The internal variables for the components of the directions of the first principal stress are named `solid.sp1x`, `solid.sp1y`, and `solid.sp1z`. The direction vectors for the other two principal stresses are named analogously.

In terms of the principal stresses, the stress invariants can be written as

$$I_1(\sigma) = \sigma_{p1} + \sigma_{p2} + \sigma_{p3}$$
$$I_2(\sigma) = \sigma_{p1}\sigma_{p2} + \sigma_{p2}\sigma_{p3} + \sigma_{p1}\sigma_{p3}$$
$$I_3(\sigma) = \sigma_{p1}\sigma_{p2}\sigma_{p3}$$

The principal stresses are the roots of the *characteristic equation* (Cayley–Hamilton theorem)

$$\sigma_p^{\,3} - I_1\sigma_p^{\,2} + I_2\sigma_p - I_3 = 0$$

The expressions described in Equation 3-10 and Equation 3-11 enter into expressions that define various kind of yield and failure surfaces. A yield surface is a surface in the 3D space of principal stresses that circumscribe an elastic state of stress.

A **Maximum Principal Tensile Stress** plot and a **Maximum Principal Compressive Stress** plot are available under **Result Templates** for relevant physics interfaces.

HAIGH–WESTERGAARD COORDINATES

It is possible to define other variables in terms of the principal invariants. One common auxiliary variable is the *Lode angle* θ.

$$\cos 3\theta = \frac{3\sqrt{3}}{2} \cdot \frac{J_3}{J_2^{3/2}} \tag{3-12}$$

The Lode angle is part of a cylindrical coordinate system called the Haigh–Westergaard coordinates. The height (in the hydrostatic axis direction) corresponds to $\xi = I_1/\sqrt{3}$ and the radius corresponds to $r = \sqrt{2J_2}$.

The Lode angle is bounded to $0 \le \theta \le \pi/3$ when the principal stresses are sorted as $\sigma_{p1} \ge \sigma_{p2} \ge \sigma_{p3}$ (Ref. 54). Following this convention, $\theta = 0$ corresponds to the *tensile meridian*, and $\theta = \pi/3$ corresponds to the *compressive meridian*.

The principal stresses (σ_{p1}, σ_{p2}, and σ_{p3}) are then written in terms of the invariants I_1, J_2, and the Lode angle θ as

$$\sigma_{p1} = \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}}\cos\theta$$
$$\sigma_{p2} = \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}}\cos\left(\theta - \frac{2\pi}{3}\right)$$
$$\sigma_{p3} = \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}}\cos\left(\theta + \frac{2\pi}{3}\right)$$

The Lode angle is undefined at the hydrostatic axis, where all three principal stresses are equal ($\sigma_{p1} = \sigma_{p2} = \sigma_{p3} = I_1/3$) and $J_2 = 0$. To avoid division by zero, the Lode angle is actually computed from the inverse tangent function `atan2`, instead of the inverse cosine, as stated in Equation 3-12.

The Lode angle and the equivalent (von Mises) stress can be called in user defined yield criteria by referencing the variables `solid.thetaL` and `solid.mises`, where `solid` is the name of the physics interface node.

OCTAHEDRAL PLANE

The *octahedral plane* (also called π-plane or deviatoric plane) lies perpendicular to the hydrostatic axis in the Haigh–Westergaard Coordinates. The stress normal to this plane

is $\sigma_{oct} = I_1/3$, also called *mean stress* or *hydrostatic stress*, and the shear stress on that plane is defined by $\tau_{oct} = \sqrt{2/3 J_2}$.

The profile of the yield surface on the octahedral plane is useful for visualizing changes in yield limits under compression or tension, and it illustrates how the criterion depends on the Lode angle. See for instance the Tresca Criterion and the Mohr–Coulomb Criterion.

OCTAHEDRAL SECTION

The profile of the yield function on the Octahedral Plane is useful for denoting changes in the yield limit under different loading conditions, for example, different yield limits in compression or tension. For Porous Plasticity, COMSOL Multiphysics provides three alternative definitions of the function $\Gamma(\theta)$ that controls the shape of the yield function in the octahedral plane:

- Circular, $\Gamma(\theta) = 1$
- Gudehus
- Lagioia–Panteghini

Gudehus (Ref. 99) proposed an octahedral section that changes with the Lode angle θ. The change in the cross-sectional shape depends on one parameter as

$$\Gamma(\theta) = \frac{1}{2}\left(1 + \cos(3\theta) + \frac{1}{c}(1 - \cos(3\theta))\right) \tag{3-13}$$

A value of $c = 1$ represents a circular section. Independently of the value of the parameter c, the function equals $\Gamma(0) = 1$ at the tensile meridian ($\theta = 0$), and $\Gamma(\pi/3) = 1/c$ at the compressive meridian. For $c < 1$, the tensile strength is lower than the compressive strength. The function $\Gamma(\theta)$ is thus an approximation of the ratio of tensile to compressive strength. For the function to be convex, the parameter c has a lower bound of $7/9$ and an upper bound of $9/7$.

Panteghini and Lagioia (Ref. 100) proposed a three-parameter function to control the octahedral section based on the Lode angle:

$$\Gamma(\theta) = c_1 \cos\left(\frac{1}{3}\operatorname{acos}(c_2 \cos(3\theta)) - c_3\frac{\pi}{6}\right) \tag{3-14}$$

The function $\Gamma(\theta)$ was derived with the purpose to recover common soil plasticity criteria, including a rounded Mohr–Coulomb Criterion, the Matsuoka–Nakai Criterion, and the Lade–Duncan Criterion. However, the three parameters provide a

flexible option to control the octahedral section in a general manner. Setting the parameter $c_2 < 1$ creates a continuous and C^2 differentiable function.

For Elastoplastic Soils, COMSOL Multiphysics provides three alternative definitions for the function $\Gamma(\theta)$:

- Circular, $\Gamma(\theta) = 1$
- Mohr–Coulomb Criterion
- Matsuoka–Nakai Criterion

MERIDIONAL PLANE

The meridional plane, also called *Rendulic plane*, is the profile of the yield criterion at a given Lode angle, typically at the compressive meridian $\theta = \pi/3$.

In the Haigh–Westergaard Coordinates, the meridional plane is given by the coordinates $\xi = I_1/\sqrt{3}$ and $r = \sqrt{2J_2}$, but it also common to visualize criteria in the pq-plane, given by the coordinates $p = I_1/3$ and $q = \sqrt{3J_2}$, or the plane given by the axes I_1 and $\sqrt{J_2}$.

Plane Stress

For two-dimensional problems, there are tree possible approximations: Plane Strain, Generalized Plane Strain, and *plane stress*. The selection is made in the settings for the Solid Mechanics node.

In the plane stress formulation in COMSOL Multiphysics, the plane stress conditions

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \tag{3-15}$$

are not enforced through a modified constitutive relation, as is common in many textbooks. Instead, an extra set of degrees of freedom for the out-of-plane strains are introduced, and Equation 3-15 is enforced by solving for the strains.

For a general anisotropic linear elastic material in case of plane stress, COMSOL Multiphysics solves three equations. For isotropy and orthotropy, only one extra

degree of freedom is needed since the out-of-plane shear components of the stress tensor are zero.

- For isotropic and orthotropic materials, the extra degree of freedom is named `wZ`, and represents $\partial w/\partial Z$.
- For anisotropic materials in 3D or 2D, two more degrees of freedom area added, `uZ` and `vZ`. They represent $\partial u/\partial Z$ and $\partial v/\partial Z$.

Initial Stress and Strain

Initial stresses and strains refer to a stress and strain state that would exist even without external loads. Initial stresses and strains are not initial values in the mathematical sense. They apply all through the solution, and may even vary with time or solution parameters. They should rather be considered as an offset to the stress and strain state in the constitutive relation.

The initial strain is subtracted from the total strain, before the constitutive law is applied for computing the stresses. The initial stress is added to the stress computed by using the constitutive law. As an example, linear elasticity including both an initial strain ε_0 and an initial stress s_0 can be written as

$$s = s_0 + \mathsf{C} : (\varepsilon - \varepsilon_0)$$

It can also be noted that the effect of the initial strain is analogous to, for example, a thermal strain or other inelastic strain contributions.

Use either the initial strain or the initial stress (but not both) when you have results from another analysis or another physics interface, which you want to incorporate into the stress–strain relation.

Both the initial stress and strain are tensor variables defined via components in the local coordinate system for each domain.

In case of nearly incompressible material (mixed formulation), the components of the total initial stress (that is, without volumetric-deviatoric split) are still input. The initial pressure in the equation for the pressure help variable p_{w} is computed as

$$p_0 = -\frac{1}{3} I_1(s_0)$$

In the case of geometric nonlinearity, the initial stress represents the second Piola–Kirchhoff stress, not the Cauchy stress. The initial strain is interpreted as a Green–Lagrange strain or Hencky strain, depending on the selection in the **Geometric Nonlinearity** section.

OTHER POSSIBLE USES OF INITIAL STRAINS AND STRESSES

Many inelastic effects in solids mechanics (for example creep, plasticity, damping, viscoelasticity, poroelasticity, and so on) are represented by additive contributions to either the total strain or total stress. Then the initial value input fields can be used for coupling the elastic equations (solid mechanics) to the *constitutive equations* (usually General Form PDEs) modeling such extra effects. When adding stress contributions, you may however find it more convenient to use the External Stress concept.

The **Initial Stress and Strain** node can be added to Linear Elastic Material, Nonlinear Elastic Materials, Piezoelectricity, Elastoplastic Soils, Magnetostriction and Piezomagnetism, or Shape Memory Alloy materials.

External Stress

The external stress represents an additional stress contribution which has a source other than the constitutive relation. It is similar to the initial stress described in Initial Stress and Strain, and the two features can be used interchangeably. As an example, linear elasticity including an external stress σ_{ext} can be written as

$$\sigma = \sigma_{ext} + C : \varepsilon \tag{3-16}$$

This additive contribution is the default option when adding external stresses to either the material representation (second Piola–Kirchhoff stress) or spatial representation (Cauchy stress).

It is also possible to prescribe a stress contribution that only acts as a load on the structure, but that is not added into the stress tensor definition as described in Equation 3-16. The typical case is when there is a pore pressure in a porous material, a common case in geotechnical engineering. The stress carried by the solid material excluding the pore pressure is often called the *effective stress*. The load from the pore pressure helps to balance the external loads, while not contributing to the stress tensor of the solid. The contribution to the virtual work of the external stress (load) is then

$$\delta W = \int_V -\delta\varepsilon : \sigma_{ext}\, dv \tag{3-17}$$

With the External Stress feature it is also possible to model residual stresses due to, for instance, manufacturing processes. The **Residual stress** option augments the stress tensor as defined in Equation 3-16, and it also removes the deformation created from it by applying an external load with opposite sign, as described in Equation 3-17. In this way there is no induced deformation in the solid, but the definition of the stress tensor and its invariants are augmented, which has an impact in plasticity, creep, or viscoplasticity.

When **External Stress (Nominal)** is selected, the external stress tensor does not have to by symmetric, thus the contribution of the external stress acts as an external load, to the virtual work is

$$\delta W = \int_V -\delta\nabla\mathbf{u} : P_{\text{ext}}\, dv$$

IN SITU STRESS

The in situ stress is a common residual stress used in geotechnical engineering. The *vertical stress*, σ_v, also called *overburden pressure*, *lithostatic pressure*, or *confining pressure*, represents the stress in a point given by the weight of the overlaying material.

$$\sigma_v = \rho g D$$

The elevation D is the distance of a point in the soil to the top boundary, ρ is the density and g the acceleration of gravity. This analytical expression for the vertical stress is derived for a slab of soil of infinite lateral extension.

The lateral or *horizontal stress* σ_h is normally given as a factor of the vertical stress

$$\sigma_h = k_0 \sigma_v$$

The factor k_0, called the *coefficient of lateral earth pressure*, is normally computed from the angle of internal friction, from the Poisson's ratio, or more complex formulas.

When the z-axis represents the vertical coordinate, the in situ stress tensor is written as

$$\sigma_{\text{ins}} = \begin{bmatrix} \sigma_h & & \\ & \sigma_h & \\ & & \sigma_v \end{bmatrix}$$

The **External Stress** subnode can be added to Linear Elastic Material, Nonlinear Elastic Materials, Elastoplastic Soils, Shape Memory Alloy, or Hyperelastic Materials.

External Strain

The external strain represents an inelastic strain contribution which has a source other than the elastic deformation. It is similar to the initial strain described in Initial Stress and Strain, and the two features can be used interchangeably. As an example, linear elasticity including an external strain ε_{ext} can be written as

$$\sigma = C : \varepsilon_{el} \text{ with } \varepsilon_{el} = \varepsilon - \varepsilon_{ext} \tag{3-18}$$

This Additive Decomposition of strains, $\varepsilon = \varepsilon_{el} + \varepsilon_{ext}$, is the default option when adding external strains to represent either the second Piola–Kirchhoff stress or the Cauchy stress.

It is possible to prescribe a large external strain contribution that follows a Multiplicative Decomposition of the deformation gradient tensor. In this case, the elastic deformation gradient reads

$$F_{el} = FF_{ext}^{-1} \tag{3-19}$$

Enter the external deformation gradient, F_{ext}, the external stretches in the local coordinate system, λ_{ext}, or the inverse deformation gradient, F_{ext}^{-1}. Is is also possible to enter the external volumetric strain, dV_{ext}, so the external volume change reads $J_{ext} = 1 + dV_{ext}$. The external strain can also be computed by Using External Materials.

The **External Strain** subnode can be added to Linear Elastic Material, Nonlinear Elastic Materials, Shape Memory Alloy, or Hyperelastic Materials.

See also

- Using External Materials
- External Material and Working with External Materials in the *COMSOL Multiphysics Reference Manual.*

Axial Symmetry and Stresses

The physical stress components are defined on the global coordinate system:

$$\sigma_r = \sigma^{11}$$

$$\sigma_{\varphi} = r^2 \sigma^{22}$$

$$\sigma_z = \sigma^{33}$$

$$\sigma_{rz} = \sigma^{13}$$

The first invariant of the stress tensor is

$$I_1(\sigma) = \sum_{i,j} (\sigma^{ij} g_{ij}) = \sigma_r + \sigma_{\varphi} + \sigma_z$$

Equation of Motion

The first Piola–Kirchhoff stress P is calculated from the second Piola–Kirchhoff stress as $P = FS$. The first Piola–Kirchhoff stress relates forces in the present configuration with areas in the reference configuration, and it is sometimes called the *nominal stress*.

Using the first Piola–Kirchhoff stress tensor, the equation of motion can be written in the following form:

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_X \cdot P^{\mathrm{T}} + \mathbf{F}_V \tag{3-20}$$

where the density corresponds to the material density in the initial undeformed state, the volume force vector $\mathbf{F}_V$ has components in the actual configuration but given with respect to the undeformed volume, and the tensor divergence operator is computed with respect to the coordinates on the material frame. Equation 3-20 is the strong form that corresponds to the weak form equations within the Solid Mechanics interface (and many related multiphysics interfaces) in COMSOL Multiphysics. Using vector and tensor components, the equation can be written as

$$\rho_0 \frac{\partial^2 u_x}{\partial t^2} = \left(\frac{\partial P_{xX}}{\partial X} + \frac{\partial P_{xY}}{\partial Y} + \frac{\partial P_{xZ}}{\partial Z}\right) + F_{Vx}$$

$$\rho_0 \frac{\partial^2 u_y}{\partial t^2} = \left(\frac{\partial P_{yX}}{\partial X} + \frac{\partial P_{yY}}{\partial Y} + \frac{\partial P_{yZ}}{\partial Z}\right) + F_{Vy}$$

$$\rho_0 \frac{\partial^2 u_z}{\partial t^2} = \left(\frac{\partial P_{zX}}{\partial X} + \frac{\partial P_{zY}}{\partial Y} + \frac{\partial P_{zZ}}{\partial Z}\right) + F_{Vz}$$

The components of the first Piola–Kirchhoff stress tensor are nonsymmetric in the general case, thus

$$P_{iJ} \neq P_{Ij}$$

because the component indices correspond to different frames.

The boundary load vector $\mathbf{F}_A$ in case of geometric nonlinearity can be related to the first Piola–Kirchhoff stress tensor via the following formula:

$$\mathbf{F}_A = P\mathbf{n}_0$$

where the normal $\mathbf{n}_0$ corresponds to the undeformed surface element. This force vector is often referred to as the *nominal traction*. In components, it can be written as

$$\begin{aligned} F_{Ax} &= P_{xX}n_X + P_{xY}n_Y + P_{xZ}n_Z \\ F_{Ay} &= P_{yX}n_X + P_{yY}n_Y + P_{yZ}n_Z \\ F_{Az} &= P_{zX}n_X + P_{zY}n_Y + P_{zZ}n_Z \end{aligned}$$

The Cauchy stress, σ, can be calculated as

$$\sigma = J^{-1}PF^T = J^{-1}FSF^T$$

The Cauchy stress is a true stress that relates forces in the present configuration (spatial frame) to areas in the present configuration, and it is a symmetric tensor. Equation 3-20 can be rewritten in terms of the Cauchy stress as

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_x \cdot \sigma + \mathbf{f}_V$$

where the density corresponds to the density in the actual deformed state, the volume force vector $\mathbf{f}_V$ has components in the actual configuration (spatial frame) given with respect to the deformed volume, and the divergence operator is computed with respect to the spatial coordinates.

The pressure is computed as

$$p = -\frac{1}{3}\text{trace}(\sigma)$$

which corresponds to the volumetric part of the Cauchy stress, and it is positive in compression. The deviatoric part is defined as

$$\sigma_\text{d} = \sigma + p\mathbf{I}$$

The second invariant of the deviatoric stress

$$J_2(\sigma) = \frac{1}{2}\sigma_\text{d}{:}\sigma_\text{d}$$

is used for the computation of von Mises (equivalent) stress

$$\sigma_{\text{mises}} = \sqrt{3J_2(\sigma)}$$

Material Models

There are many material models available for structural analysis in COMSOL Multiphysics. In this section, the theory for all material models is presented. The material models available, and also some detailed aspects of them, depend on which licenses you have. The material models, grouped by families, are as follows:

- Linear Elastic Material
- Nonlinear Elastic Materials
  - Ramberg–Osgood
  - Power Law
  - Bilinear Elastic
  - Uniaxial Data
  - Shear Data
  - Hyperbolic Law
  - Hardin–Drnevich
  - Duncan–Chang
  - Duncan–Selig
  - Small-Strain Overlay
  - User Defined
- Linear Viscoelasticity
  - The Maxwell Model
  - The Generalized Maxwell Model
  - The Kelvin–Voigt Model
  - The Generalized Kelvin–Voigt Model
  - Standard Linear Solid Model
  - The Burgers Model
  - Generalized Maxwell Model with Fractional Derivatives
  - Generalized Kelvin–Voigt Model with Fractional Derivatives
  - Standard Linear Solid Model with Fractional Derivatives
  - Burgers Model with Fractional Derivatives
  - User Defined

- Hyperelastic Materials
  - Neo-Hookean
  - Saint-Venant–Kirchhoff
  - Mooney–Rivlin, Two Parameters
  - Mooney–Rivlin, Five Parameters
  - Mooney–Rivlin, Nine Parameters
  - Yeoh
  - Ogden
  - Storakers
  - Varga
  - Arruda–Boyce
  - Gent
  - van der Waals
  - Blatz–Ko
  - Gao
  - Murnaghan
  - Delfino
  - Fung
  - Extended Tube
  - User Defined
  - Mullins Effect
- Distributed Fiber Models
- Elastoplastic Materials
- Metal Plasticity
  - von Mises Criterion
  - Tresca Criterion
  - Hill Criterion

- Pressure-Dependent and Soil Plasticity
  - Drucker–Prager Criterion
  - Elliptic Criterion
  - Parabolic Criterion
  - Mohr–Coulomb Criterion
  - Drucker–Prager Criterion
  - Matsuoka–Nakai Criterion
  - Lade–Duncan Criterion
  - Compression Cap
  - Tension Cutoff
- Porous Plasticity
  - Shima–Oyane Criterion
  - Gurson Criterion
  - Gurson–Tvergaard–Needleman Criterion
  - Fleck–Kuhn–McMeeking Criterion
  - FKM–GTN Criterion
  - Capped Drucker–Prager
- Failure Criteria for Concrete and Rocks
  - Bresler–Pister Criterion
  - Willam–Warnke Criterion
  - Ottosen Criterion
  - Original Hoek–Brown Criterion
  - Generalized Hoek–Brown Criterion
- Damage Models
- Coupled Damage–Plasticity
- Elastoplastic Soils
  - The Modified Cam-Clay Soil Model
  - The Modified Structured Cam-Clay Soil Model
  - The Extended Barcelona Basic Soil Model
  - The Hardening Soil Model
  - The Hardening Soil Small Strain Model

- Creep and Viscoplasticity
  - Creep
  - Viscoplasticity
  - Creep and Viscoplasticity for Large Strains
- Inelastic Strain Rate
- Shape Memory Alloy
- Piezoelectricity
- Magnetomechanics
- Magnetostriction and Piezomagnetism
- Electrostriction
- Ferroelectroelasticity
- Rigid Material
- Safety Factor Evaluation

You can also add a material model which you have coded yourself and made available as a binary library file using an External Stress–Strain Relation.

In the *COMSOL Multiphysics Reference Manual*:

- Working with External Materials
- External Material

Linear Elastic Material

For a linear elastic material, Hooke's law relates the stress tensor to the elastic strain tensor:

$$\sigma = \sigma_{ex} + \mathsf{C} : \varepsilon_{el} = \sigma_{ex} + \mathsf{C} : (\varepsilon - \varepsilon_{inel}) \quad (3\text{-}21)$$

where C is the 4th order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction). The elastic strain ε_{el} is the difference between the total strain ε and all inelastic strains ε_{inel}. There may also be an extra stress contribution σ_{ex} with contributions from initial stresses and viscoelastic stresses. In case of geometric nonlinearity, the second Piola–Kirchhoff stress tensor and the Green–Lagrange strain tensor are used.

The elastic strain energy density is

$$W_s = \frac{1}{2}\varepsilon_{el} : (\mathbf{C} : \varepsilon_{el} + 2\sigma_0) = \frac{1}{2}\varepsilon_{el} : (\sigma + \sigma_0) \tag{3-22}$$

This expression assumes that the initial stress contribution is constant during the straining of the material.

TENSOR VS. MATRIX FORMULATIONS

Because of the symmetry, the strain tensor can be written as the following matrix:

$$\begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix}$$

A similar representation applies to the stress tensor:

$$\begin{bmatrix} \sigma_x & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z \end{bmatrix}$$

Due to the symmetry, the elasticity tensor can be completely represented by a symmetric 6-by-6 matrix as

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\ D_{12} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{13} & D_{23} & D_{33} & D_{34} & D_{35} & D_{36} \\ D_{14} & D_{24} & D_{34} & D_{44} & D_{45} & D_{46} \\ D_{15} & D_{25} & D_{35} & D_{45} & D_{55} & D_{56} \\ D_{16} & D_{26} & D_{36} & D_{46} & D_{56} & D_{66} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^{1111} & \mathbf{C}^{1122} & \mathbf{C}^{1133} & \mathbf{C}^{1112} & \mathbf{C}^{1123} & \mathbf{C}^{1113} \\ \mathbf{C}^{1122} & \mathbf{C}^{2222} & \mathbf{C}^{2233} & \mathbf{C}^{2212} & \mathbf{C}^{2223} & \mathbf{C}^{2213} \\ \mathbf{C}^{1133} & \mathbf{C}^{2233} & \mathbf{C}^{3333} & \mathbf{C}^{3312} & \mathbf{C}^{3323} & \mathbf{C}^{3313} \\ \mathbf{C}^{1112} & \mathbf{C}^{2212} & \mathbf{C}^{3312} & \mathbf{C}^{1212} & \mathbf{C}^{1223} & \mathbf{C}^{1213} \\ \mathbf{C}^{1123} & \mathbf{C}^{2223} & \mathbf{C}^{3323} & \mathbf{C}^{1223} & \mathbf{C}^{2323} & \mathbf{C}^{2313} \\ \mathbf{C}^{1113} & \mathbf{C}^{2213} & \mathbf{C}^{3313} & \mathbf{C}^{1213} & \mathbf{C}^{2313} & \mathbf{C}^{1313} \end{bmatrix}$$

which is the *elasticity matrix.*

ISOTROPIC MATERIAL AND ELASTIC MODULI

In this case, the elasticity matrix becomes

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{3-23}$$

Different pairs of elastic moduli can be used, and as long as two moduli are defined, the others can be computed according to Table 3-1.

TABLE 3-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Young's modulus	E =	E	E	$\frac{9KG}{3K+G}$	$\mu\frac{3\lambda+2\mu}{\lambda+\mu}$
Poisson's ratio	ν =	ν	$\frac{E}{2G}-1$	$\frac{1}{2}\frac{3K-2G}{3K+G}$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	K =	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	K	$\lambda+\frac{2\mu}{3}$
Shear modulus	G =	$\frac{E}{2(1+\nu)}$	G	G	μ
Lamé parameter λ	λ =	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K-\frac{2G}{3}$	λ
Lamé parameter μ	μ =	$\frac{E}{2(1+\nu)}$	G	G	μ
Pressure-wave speed	c_p =	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\frac{K+4G/3}{\rho}}$	$\sqrt{\frac{\lambda+2\mu}{\rho}}$
Shear-wave speed	c_s =	$\sqrt{\frac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

According to Table 3-1, the elasticity matrix D for isotropic materials is written in terms of Lamé parameters λ and μ,

$$D = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{3-24}$$

or in terms of the bulk modulus K and shear modulus G:

$$D = \begin{bmatrix} K+\frac{4G}{3} & K-\frac{2G}{3} & K-\frac{2G}{3} & 0 & 0 & 0 \\ K-\frac{2G}{3} & K+\frac{4G}{3} & K-\frac{2G}{3} & 0 & 0 & 0 \\ K-\frac{2G}{3} & K-\frac{2G}{3} & K+\frac{4G}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \tag{3-25}$$

ORTHOTROPIC AND ANISOTROPIC MATERIALS

There are two different ways to represent orthotropic or anisotropic data. The **Standard (11, 22, 33, 12, 23, 13) material data ordering** converts the indices as

$$\begin{bmatrix} 11 \\ 22 \\ 33 \\ 12, 21 \\ 23, 32 \\ 13, 31 \end{bmatrix} \leftrightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \leftrightarrow \begin{bmatrix} x \\ y \\ z \\ xy \\ yz \\ xz \end{bmatrix}$$

thus, Hooke's law is presented in the form involving the elasticity matrix D and the following vectors:

$$\begin{bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \sigma_{xy}\\ \sigma_{yz}\\ \sigma_{xz}\end{bmatrix} = \begin{bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \sigma_{xy}\\ \sigma_{yz}\\ \sigma_{xz}\end{bmatrix}_{\text{ex}} + D\left(\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ 2\varepsilon_{xy}\\ 2\varepsilon_{yz}\\ 2\varepsilon_{xz}\end{bmatrix} - \begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ 2\varepsilon_{xy}\\ 2\varepsilon_{yz}\\ 2\varepsilon_{xz}\end{bmatrix}_{\text{inel}}\right)$$

COMSOL Multiphysics uses the complete tensor representation internally to perform the coordinate system transformations correctly.

Beside the **Standard (11, 22, 33, 12, 23, 13) Material data ordering**, the elasticity coefficients can be entered following the **Voigt** notation. In the **Voigt (11, 22, 33, 23, 13, 12) Material data ordering**, the sorting of indices is

$$\begin{bmatrix}11\\ 22\\ 33\\ 23,\,32\\ 13,\,31\\ 12,\,21\end{bmatrix} \leftrightarrow \begin{bmatrix}1\\ 2\\ 3\\ 4\\ 5\\ 6\end{bmatrix} \leftrightarrow \begin{bmatrix}x\\ y\\ z\\ yz\\ xz\\ xy\end{bmatrix}$$

The last three rows and columns in the elasticity matrix D are thus swapped.

Orthotropic Material

The elasticity matrix for orthotropic materials in the **Standard (11, 22, 33, 12, 23, 13) Material data ordering** has the following structure:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0\\ D_{12} & D_{22} & D_{23} & 0 & 0 & 0\\ D_{13} & D_{23} & D_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & D_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & D_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & D_{66}\end{bmatrix} \tag{3-26}$$

where the components are as follows:

$$D_{11} = \frac{E_x^2(E_z\nu_{yz}^2 - E_y)}{D_{\text{denom}}}, \qquad D_{12} = -\frac{E_xE_y(E_z\nu_{yz}\nu_{xz} + E_y\nu_{xy})}{D_{\text{denom}}}$$

$$D_{13} = -\frac{E_x E_y E_z(\nu_{xy}\nu_{yz} + \nu_{xz})}{D_{\text{denom}}}, \qquad D_{22} = \frac{E_y^2(E_z\nu_{xz}^2 - E_x)}{D_{\text{denom}}}$$

$$D_{23} = -\frac{E_y E_z(E_y\nu_{xy}\nu_{xz} + E_x\nu_{yz})}{D_{\text{denom}}}, \qquad D_{33} = \frac{E_y E_z(E_y\nu_{xy}^2 - E_x)}{D_{\text{denom}}}$$

$$D_{44} = G_{xy}, \quad D_{55} = G_{yz}, \text{ and } D_{66} = G_{xz}$$

where

$$D_{\text{denom}} = E_y E_z \nu_{xz}^2 - E_x E_y + 2\nu_{xy}\nu_{yz}\nu_{xz} E_y E_z + E_x E_z \nu_{yz}^2 + E_y^2 \nu_{xy}^2$$

The values of E_x, E_y, E_z, ν_{xy}, ν_{yz}, ν_{xz}, G_{xy}, G_{yz}, and G_{xz} are supplied in designated fields in the physics interface. COMSOL Multiphysics deduces the remaining components — ν_{yx}, ν_{zx}, and ν_{zy} — using the fact that the matrices D and D^{-1} are symmetric. The *compliance matrix* has the following form:

$$D^{-1} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & -\frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} \end{bmatrix}$$

!	The values of ν_{xy} and ν_{yx} are different for an orthotropic material. For a certain set of given material data, you must make sure that the definition of the indices is consistent with the definition used in COMSOL Multiphysics.

The elasticity matrix in the **Voigt (11, 22, 33, 23, 13, 12) Material data ordering** changes the sorting of the last three elements in the elasticity matrix:

$$D_{44} = G_{yz}, \quad D_{55} = G_{xz}, \text{ and } D_{66} = G_{xy}$$

If a pair of elastic moduli is present in the material definition, the values of E_x, E_y, E_z, ν_{xy}, ν_{yz}, ν_{xz}, G_{xy}, G_{yz}, and G_{xz} are computed automatically. Note that the resulting elasticity matrix will be isotropic. Depending on which pair of elastic moduli that is available, the expressions in Table 3-1 are used to find the above values.

Transversely Isotropic Material

Transversely isotropic material is a subclass of orthotropic materials, for which one principal direction in the material is different from the other two that are equivalent. This special direction is assumed to be the first axis of the selected coordinate system. Because of the symmetry, the following relations hold:

$$E_3 = E_2$$

$$\nu_{13} = \nu_{12}$$

$$G_{23} = \frac{E_2}{2(1+\nu_{23})}$$

$$G_{12} = G_{13}$$

Thus, only five elasticity parameters are needed to characterize the material.

Anisotropic Material

In the general case of fully anisotropic material, you provide explicitly all 21 components of the symmetric elasticity matrix D, in either **Standard (11, 22, 33, 12, 23, 13)** or **Voigt (11, 22, 33, 23, 13, 12) Material data ordering**.

If a pair of elastic moduli is present in the material definition, the components of the symmetric elasticity matrix D are computed using one of Equation 3-23 to Equation 3-25. Depending on which pair of elastic moduli that is available, the expressions in Table 3-1 are used to compute the necessary values. In case the orthotropic properties E_i, ν_{ij}, and G_{ij} are present in the material definition, the components of the symmetric elasticity matrix D are computed using Equation 3-26. Note that the resulting elasticity matrix will not be fully anisotropic in either case.

Crystal Symmetry

Here, the components of the elasticity matrix in Voigt notation (denoted by c_{ij}) are referred to as elasticity constants. Because of the material symmetry, only certain

components need to be specified. The following **Crystal systems** are available in COMSOL Multiphysics:

Cubic (3 constants) c_{11}, c_{12}, c_{44}

$$\begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{12} & c_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{44} \end{bmatrix}$$

This crystal system includes the following crystal classes: $23, \overline{m}3, 432, \overline{4}3m, m\overline{3}m$.

Hexagonal (5 constants) $c_{11}, c_{12}, c_{13}, c_{33}, c_{44}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{13} & 0 & 0 & 0 \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix}$$

where $c_{66} = 0{,}5(c_{11} - c_{12})$.

This crystal system includes the following crystal classes:
$6, \overline{6}, 6/m, 622, 6mm, \overline{6}2m, 6/mmm$, and it is equivalent to a Transversely Isotropic Material.

Trigonal (6 constants) $c_{11}, c_{12}, c_{13}, c_{14}, c_{33}, c_{44}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & 0 & 0 \\ c_{12} & c_{11} & c_{13} & -c_{14} & 0 & 0 \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ c_{14} & -c_{14} & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & c_{14} \\ 0 & 0 & 0 & 0 & c_{14} & c_{66} \end{bmatrix}$$

where $c_{66} = 0{,}5(c_{11} - c_{12})$.

This crystal system includes the following crystal classes: $32, 3m, \bar{3}m$.

Trigonal (7 constants) $c_{11}, c_{12}, c_{13}, c_{14}, c_{25}, c_{33}, c_{44}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & -c_{25} & 0 \\ c_{12} & c_{11} & c_{13} & -c_{14} & c_{25} & 0 \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ c_{14} & -c_{14} & 0 & c_{44} & 0 & c_{25} \\ -c_{25} & c_{25} & 0 & 0 & c_{44} & c_{14} \\ 0 & 0 & 0 & c_{25} & c_{14} & c_{66} \end{bmatrix}$$

where $c_{66} = 0{,}5(c_{11} - c_{12})$.

This crystal system includes the following crystal classes: $3, \bar{3}$.

Tetragonal (6 constants) $c_{11}, c_{12}, c_{13}, c_{33}, c_{44}, c_{66}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{13} & 0 & 0 & 0 \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix}$$

This crystal system includes the following crystal classes: $422, 4mm, \bar{4}3m, 4/mmm$.

Tetragonal (7 constants) $c_{11}, c_{12}, c_{13}, c_{16}, c_{33}, c_{44}, c_{66}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & c_{16} \\ c_{12} & c_{11} & c_{13} & 0 & 0 & -c_{16} \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ c_{16} & -c_{16} & 0 & 0 & 0 & c_{66} \end{bmatrix}$$

This crystal system includes the following crystal classes: $4, \bar{4}, 4/m$.

Orthorhombic (9 constants) $c_{11}, c_{12}, c_{13}, c_{22}, c_{23}, c_{33}, c_{44}, c_{55}, c_{66}$

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix}$$

This crystal system includes the following crystal classes: $222, mm2, mmm$. This type of crystal symmetry is equivalent to an Orthotropic Material.

AXIAL SYMMETRY

For the linear elastic material, the stress components in coordinate system are

$$\sigma^{ij} = \sigma^{ij}_{\text{ex}} + \mathsf{C}^{ijkl}(\varepsilon_{kl} - \varepsilon_{\text{inel},kl})$$

For anisotropic and orthotropic materials, the 4th-order elasticity tensor is defined from the D matrix according to:

$$\begin{bmatrix} \sigma_r \\ \sigma_\varphi \\ \sigma_z \\ \sigma_{rz} \end{bmatrix} = \begin{bmatrix} \sigma_r \\ \sigma_\varphi \\ \sigma_z \\ \sigma_{rz} \end{bmatrix}_{\text{ex}} + D\left(\begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_z \\ 2\varepsilon_{rz} \end{bmatrix} - \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_z \\ 2\varepsilon_{rz} \end{bmatrix}_{\text{inel}} \right)$$

The user input D matrix always contains the physical components of the elasticity tensor

$$\mathsf{C}^{\text{phys}}_{ijkl}$$

and the corresponding tensor components are computed internally according to:

$$\mathsf{C}^{ijkl} = \frac{\mathsf{C}^{\text{phys}}_{ijkl}}{\sqrt{g_{ii}}\sqrt{g_{jj}}\sqrt{g_{kk}}\sqrt{g_{ll}}}$$

For an isotropic material:

$$\mathsf{C}^{ijkl} = \lambda g^{ij} g^{kl} + \mu(g^{ik} g^{jl} + g^{il} g^{jk})$$

where λ and μ are the first and second Lamé elastic parameters and g is the metric tensor.

For a hyperelastic material, the second Piola–Kirchhoff stress tensor is computed as

$$S = 2\frac{\partial W_s}{\partial C}$$

which is computed as the contravariant components of the stress in the local coordinate system:

$$S^{ij} = 2\frac{\partial W_s}{\partial C_{ij}}$$

The energy variation is computed as

$$S : \text{test}(\varepsilon) = \sum_{i,j} S^{ij}\text{test}(\varepsilon_{ij})$$

which can be also written as

$$S_r\text{test}(\varepsilon_r) + S_\varphi\text{test}(\varepsilon_\varphi) + S_z\text{test}(\varepsilon_z) + 2S_{rz}\text{test}(\varepsilon_{rz})$$

ENTROPY AND THERMOELASTICITY

The free energy for the linear thermoelastic material can be written as

$$F = \rho f_0(T) + W_s(\varepsilon, T)$$

where the strain energy density $W_s(\varepsilon, T)$ is given by Equation 3-22. Hence, the stress can be found as

$$\sigma = \left(\frac{\partial F}{\partial \varepsilon}\right)_T = \left(\frac{\partial W}{\partial \varepsilon}\right)_T = C : (\varepsilon - \varepsilon_{\text{inel}})$$

and the entropy per reference volume can be calculated as

$$-\left(\frac{\partial F}{\partial T}\right)_\varepsilon = \rho C_p \log(T/T_0) + S_{\text{elast}}$$

where T_0 is a reference temperature, the volumetric heat capacity ρC_p can be assumed to be independent of the temperature (Dulong–Petit law), and the elastic entropy is

$$S_{\text{elast}} = \alpha : \sigma$$

where α is the thermal expansion coefficient tensor. For an isotropic material, it simplifies into

$$S_{\text{elast}} = \alpha(\sigma_x + \sigma_y + \sigma_z)$$

The heat balance equation can be written as

$$\rho C_{\text{p}} \frac{\partial T}{\partial t} + T \frac{\partial}{\partial t} S_{\text{elast}} = \nabla \cdot (k \nabla T) + Q_{\text{h}}$$

where k is the thermal conductivity matrix, and the heat source caused by the dissipation is

$$Q_{\text{h}} = \tau : \dot{\varepsilon}$$

where $\dot{\varepsilon}$ is the strain-rate tensor and the tensor τ represents all possible inelastic stresses (for example, a viscous stress).

Using the tensor components, the heat balance can be rewritten as

$$\rho C_{\text{p}} \frac{\partial T}{\partial t} + \sum_{m,\,n} T \alpha_{mn} \frac{\partial}{\partial t} \sigma_{mn} = \nabla \cdot (k \nabla T) + Q_{\text{h}} \tag{3-27}$$

In many cases, the second term can be neglected in the left-hand side of Equation 3-27 because all $T\alpha_{mn}$ are small. The resulting approximation is often called uncoupled thermoelasticity.

WAVE SPEED COMPUTATION

In case of geometric linearity, the governing equations for a linear elastic medium of any anisotropy can be written in terms of the structural displacement vector $\mathbf{u}$ as

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot (\mathsf{C} : \nabla \mathbf{u})$$

where C is the elasticity tensor.

Since the equations are linear, they possess the following time-harmonic wave solutions:

$$\mathbf{u} = \mathbf{u}_n e^{i\omega t - i\mathbf{k}\cdot\mathbf{r}}$$

where $\mathbf{k} = k\mathbf{n}$ is the wave number vector, and $\mathbf{n}$ is the direction vector that defines the wavefront propagation direction. The wavefront is an imaginary line connecting solid particles of the same phase. The velocity of such wavefront in the direction normal to it is given by the phase velocity $c = \omega/k$.

Using such a wave solution form leads to Christoffel's equation:

$$\rho c^2 \mathbf{u}_n = \Gamma_n \cdot \mathbf{u}_n \tag{3-28}$$

where the Christoffel's tensor is defined as

$$\Gamma_n = \mathbf{n} \cdot \mathsf{C} \cdot \mathbf{n} \quad .$$

The Christoffel's equation can be considered as an eigenvalue problem. Thus, to have a nontrivial solution $\mathbf{u}_n$, the phase velocity must satisfy

$$\det(\Gamma_n/\rho - c^2 I) = 0$$

which is often called the dispersion relation. In a general case, this is a cubic polynomial with three roots $c^2 = c_j^2(\Gamma_n/\rho)$. Thus, for an arbitrary anisotropic medium, three waves with different phase velocities can propagate in each given direction.

If the wave propagation is initiated by a small perturbation that is initially localized in space, the solution can be found using the Fourier and Laplace transforms, and it will represent a so-called wave packet. The wave packet will propagate with the group velocity given by

$$\frac{\partial \omega_j}{\partial \mathbf{k}} = \frac{1}{2\rho c_j} \frac{\partial}{\partial \mathbf{n}} (\mathbf{u}_{n,j} \cdot \Gamma_n \cdot \mathbf{u}_{n,j}), \quad j = 1, 2, 3$$

where $\mathbf{u}_{n,j}$ is the wave polarization vector that is the eigenvector corresponding to the eigenvalue solution $c_j^2(\Gamma_n/\rho)$ of Christoffel's equation.

COMSOL Multiphysics provides predefined variables for the phase and group velocities for waves of different types propagating in any chosen direction. These

variables do not affect the solution as such, but are available during result presentation if the Wave Speeds node has been added to the material.

The wave speed variables can be found in the **Wave speeds** folder under **Solid Mechanics** in the **Replace Expression** tree.

Mixed Formulation

Nearly incompressible materials can cause numerical problems if only displacements are used in the interpolating functions. Small errors in the evaluation of the volumetric strain, due to the finite resolution of the discrete model, are exaggerated by the high bulk modulus (or low bulk modulus to shear modulus ratios). This leads to an unstable representation of stresses, and in general, to an underestimation of the deformation, as spurious volumetric stresses might balance applied shear and bending loads.

When the **Pressure formulation** is selected in the **Use mixed formulation** list, the volumetric stress p_w is used as an additional dependent variable. The resulting mixed formulation is also known as a *u-p formulation*. This formulation removes the effect of the volumetric strain from the original stress tensor, and replaces it with an interpolated pressure, p_w. A separate equation constrains the auxiliary pressure variable to make it equal (in an average sense) to the original pressure which is calculated from the strains and material model.

When the **Strain formulation** is selected in the **Use mixed formulation** list, the volumetric strain ε_w is instead used as the additional dependent variable.

The mixed formulation is beneficial when the material data is such that the deformation is close to being incompressible. For an isotropic elastic material, this happens when the Poisson's ratio approaches 0.5.

The mixed formulation is useful not only for linear elastic materials but also for nonlinear elastic materials, elastoplastic materials, hyperelastic materials, and viscoelastic materials.

The order of the shape function for the auxiliary variable (p_w or ε_w) should be of lower order than that of the displacement field. Note that some iterative solvers do not work well together with *mixed formulation* because the stiffness matrix becomes indefinite.

When the **Pressure formulation** is selected for isotropic linear elastic materials, the stress tensor s, computed directly from the strains, is replaced by a modified version:

$$\tilde{s} = s + (p - p_w)\mathbf{I}$$

where $\mathbf{I}$ is the unit tensor. The pressure p is calculated from the stress tensor as

$$p = -\frac{1}{3}\text{trace}(s)$$

This is equivalent to define

$$\tilde{s} = \text{dev}(s) - p_w\mathbf{I}$$

The auxiliary dependent variable p_w is set equal to p using the equation

$$\frac{p_w - p}{K} = 0 \tag{3-29}$$

where K is the bulk modulus. Scaling by the bulk modulus is necessary, since typical values for the auxiliary pressure p_w are in the order of 10^6 to 10^9 Pa, while typical values for the displacement degrees of freedom are orders of magnitude smaller.

The modified stress tensor $\tilde{s}$ is then used then in calculations of the energy variation.

When the **Strain formulation** is selected for isotropic linear elastic materials, the auxiliary volumetric strain ε_w is used instead of the auxiliary pressure p_w, and it is the set equal to the volumetric strain ε_{vol} using the equation

$$-K(\varepsilon_w - \varepsilon_{vol}) = 0 \tag{3-30}$$

The modified stress tensor then reads

$$\tilde{s} = s + K(\varepsilon_w - \varepsilon_{vol})\mathbf{I} = \text{dev}(s) + K\varepsilon_w\mathbf{I}$$

The advantage of using the **Strain formulation** is that the values for the auxiliary strain ε_w often are of a similar order of magnitude as the displacement degree of freedom.

For orthotropic and anisotropic materials, the auxiliary pressure equation is scaled to make the stiffness matrix symmetric. Note, however, that the stiffness matrix in this formulation is not positive definite and even contains a zero block on the diagonal in

the incompressible limit. This limits the possible choices of direct and iterative linear solver.

In case of linear elastic materials with geometric nonlinearity (and also for hyperelastic materials), the stress tensor s in the above equations is replaced by the second Piola–Kirchhoff stress tensor $\boldsymbol{S}$, see Nearly Incompressible Hyperelastic Materials.

For Plane Stress problems (in 2D solid mechanics, shell, or membrane), it is also possible to use an implicit incompressibility formulation. Based on the plane stress equation (the normal stress is equal to zero), the pressure is analytically found and the incompressibility constraint is enforced using the auxiliary transverse strain degree of freedom.

Nonlinear Elastic Materials

As opposed to hyperelastic materials, where the stress–strain relationship becomes significantly nonlinear at moderate to large strains, nonlinear elastic materials present nonlinear stress–strain relationships even at infinitesimal strains.

Here, nonlinear effects on the strain tensor are not as relevant as the nonlinearity of the elastic properties. Important materials of this class are Ramberg–Osgood for modeling metal and other ductile materials, and the Duncan–Chang soil model.

The nonlinear elastic materials as such do not include strain-rate nor stress-rate in the constitutive equations. It is however possible to add linear viscoelasticity to these materials.

For a nonlinear material to be "energetically sound" it should be possible to take any path in stress–strain space and return to the undeformed state without producing or dissipating any net energy. A requirement is then that the bulk modulus depends only on the volumetric strain, and the shear modulus depends only on the shear strains.

The splitting into volumetric and deviatoric components of the stress tensor helps ensuring the "path independent" restriction for isotropic nonlinear elastic materials.

For isotropic linear elastic materials, the stress tensor follows Hooke's law:

$$\sigma = \sigma_{ex} + C : \varepsilon_{el} = \sigma_{ex} + C : (\varepsilon - \varepsilon_{inel})$$

For a more detailed discussion, see Equation 3-21.

It is possible to split the stress and elastic strain tensors into the deviatoric and volumetric contributions

$$\sigma = \text{dev}(\sigma) + \frac{1}{3}\text{trace}(\sigma)I$$

and

$$\varepsilon_{el} = \text{dev}(\varepsilon_{el}) + \frac{1}{3}\text{trace}(\varepsilon_{el})I$$

Assuming only elastic stresses in linear isotropic elastic medium, Hooke's law simplifies to

$$\sigma = \text{dev}(\sigma) + \frac{1}{3}\text{trace}(\sigma)I = 2G\text{dev}(\varepsilon_{el}) + K\text{trace}(\varepsilon_{el})I$$

where K is the bulk modulus and G is the shear modulus. By using the convention that the pressure is the mean stress defined as positive in compression,

$$p = -\frac{1}{3}\text{trace}(\sigma)$$

The volumetric strain (positive in tension) is

$$\varepsilon_{el,vol} = \text{trace}(\varepsilon_{el})$$

The linear relation between pressure and volumetric elastic strain is thus

$$p = -K\varepsilon_{el,vol}$$

The deviatoric stress and deviatoric elastic strain tensors are related by the shear modulus

$$\text{dev}(\sigma) = 2G\text{dev}(\varepsilon_{el})$$

By using the contraction of the deviatoric stress and strain tensors, the invariants of these tensors can alternatively be related through

$$J_2(\sigma) = \frac{1}{2}\text{dev}(\sigma):\text{dev}(\sigma) = (2G)^2 J_2(\varepsilon_{el})$$

For a body subject to pure torsion on the plane 12, the stress tensor components are zero except the shear stress $\sigma_{12} = \sigma_{21} = \tau$, and also the elastic strain tensor has zero components beside the shear strains on that plane $\varepsilon_{12} = \varepsilon_{21} = \gamma_{el}/2$.

Then

$$J_2(\sigma) = \frac{1}{2}\text{dev}(\sigma):\text{dev}(\sigma) = \tau^2$$

and

$$J_2(\varepsilon_{el}) = \frac{1}{2}\text{dev}(\varepsilon_{el}):\text{dev}(\varepsilon_{el}) = \frac{1}{4}\gamma_{el}^2$$

The shear stress on the plane $\tau = \sqrt{J_2(\sigma)}$ is then related to the elastic shear strain $\gamma_{el} = 2\varepsilon_{12} = 2\sqrt{J_2(\varepsilon_{el})}$ by the shear modulus

$$\tau = G\gamma_{el}$$

Nonlinear Moduli

For nonlinear elastic materials, there is a nonlinear relation between shear stress and shear strain and/or a nonlinear relation between pressure and volumetric strain.

For the purpose of this discussion, $\tau = \sqrt{J_2(\sigma)}$ and $\gamma_{el} = 2\sqrt{J_2(\varepsilon_{el})}$ are used alternatively as variables.

In the most general case:

$$p = p(\varepsilon_{el,vol}) \text{ and } \tau = \tau(\gamma_{el})$$

Tangent and Secant Moduli

The tangent shear modulus $G_t(\gamma_{el})$ and the secant shear modulus $G_s(\gamma_{el})$ in the most general case depend nonlinearly on the shear strain, and are defined as

$$G_t = \frac{\partial\tau}{\partial\gamma_{el}} \text{ and } G_s = \frac{\tau}{\gamma_{el}}$$

Note that the secant modulus is sometimes called the ***chord modulus*** between zero and current strain level.

The tangent bulk modulus $K_t(\varepsilon_{el,vol})$ and the secant bulk modulus $K_s(\varepsilon_{el,vol})$ depend on the elastic volumetric strain, and are defined as

$$K_t = -\frac{\partial p}{\partial\varepsilon_{el,vol}} \text{ and } K_s = -\frac{p}{\varepsilon_{el,vol}}$$

For linear elastic materials, it is clear that $G_t = G_s = G$ and $K_t = K_s = K$, but this is not the case for nonlinear elastic materials.

At zero strain, the secant and shear moduli are equal to each other $G_s(0) = G_t(0)$ and $K_s(0) = K_t(0)$.

The nonlinear elastic materials described in the next sections are represented by introducing nonlinear secant shear and/or bulk moduli.

Geometric Nonlinearity

The nonlinear elastic material models are primarily intended for small-strain analysis. When used in a geometrically nonlinear study step, the strains will be interpreted as Green–Lagrange strains and the stresses will be interpreted as second Piola–Kirchhoff stresses. This is relevant for a situation with large rotations but small strains. If the strains become larger than a few percent, then you must be careful when interpreting input parameters and results since the strain and stress tensors also have a nonlinear dependence on the displacements.

RAMBERG–OSGOOD

The Ramberg–Osgood material model (Ref. 85) is a nonlinear elastic material commonly used to model plastic deformation in metals, but it also often used in soil engineering. As it is an elastic model, it can only represent plasticity during pure on-loading conditions.

For uniaxial extension, the stress–strain curve is defined by the expression

$$\varepsilon = \frac{\sigma}{E} + \varepsilon_{ref}\left(\frac{\sigma}{\sigma_{ref}}\right)^n$$

Here, E means the initial Young's modulus, and ε_{ref} is the strain at a reference stress σ_{ref}. The parameter n is the stress exponent. It is common to use $\varepsilon_{ref} = 0.002$, so σ_{ref} is the stress at 0.2% strain, typically denoted by the symbol $\sigma_{0.2}$. This parameter has several names depending on the literature: 0.2% offset yield strength, 0.2% proof stress, 0.2% proof strength, or 0.2% yield stress. Typical values for stainless steel are $E = 200$ GPa, $\sigma_{0.2} = 600$ MPa, and $n = 4.8$.

The linear strain is given by

$$\varepsilon_{el} = \frac{\sigma}{E}$$

and the nonlinear strain by

$$\varepsilon_{nl} = \varepsilon_{ref}\left(\frac{\sigma}{\sigma_{ref}}\right)^n$$

The total strain is the sum of linear and nonlinear strains

$$\varepsilon = \varepsilon_{el} + \varepsilon_{nl} = \frac{\sigma}{E} + \varepsilon_{ref}\left(\frac{\sigma}{\sigma_{ref}}\right)^n$$

In order to avoid a circular dependence of internal variables, the nonlinear strain ε_{nl} is defined with an auxiliary degree of freedom, so the stress reads $\sigma = E(\varepsilon - \varepsilon_{nl})$.

Ramberg–Osgood Material in Soil Engineering

In soil engineering, it is common to write the Ramberg–Osgood material with the stress–strain expression

$$\varepsilon = \frac{\sigma}{E} + \alpha\frac{\sigma_{ref}}{E}\left(\frac{\sigma}{\sigma_{ref}}\right)^n \tag{3-31}$$

so at the reference stress σ_{ref}, the strain is $\varepsilon = (1 + \alpha)\sigma_{ref}/E$. It is common to use $\alpha = 3/7$, so σ_{ref} represents the stress level at which the secant Young's modulus has been reduced to 70% its initial value: $E_{0.7} = E/(1 + \alpha) = 0.7E$. At this reference stress the strain is $\varepsilon = \sigma_{ref}/E_{0.7}$.

POWER LAW

For this type of material, the shear stress is related to the elastic shear strain γ by the strain exponent n and a reference shear strain γ_{ref} (Ref. 86)

$$\tau = G_0\gamma_{ref}\left(\frac{\gamma}{\gamma_{ref}}\right)^n$$

The secant shear modulus is given by the power law relation

$$G_s = G_0\left(\frac{\gamma}{\gamma_{ref}}\right)^{n-1}$$

The strain exponent controls the nonlinear deformation:

- For $n > 1$ the material behaves as a dilatant (shear-thickening) solid
- For $n = 1$ the material is linear elastic
- For $0 < n < 1$ the material behaves as pseudoplastic (shear-thinning) solid
- For $n = 0$ the material is perfectly plastic

BILINEAR ELASTIC

The most commonly mentioned model of "bilinear elastic" material is defined with two different bulk moduli for either tension and compression. Commonly, brittle

materials like graphite and ceramics exhibit this behavior. The secant bulk modulus reads

$$K = K_t \text{ for } \varepsilon_{el,vol} > 0$$

and

$$K = K_c \text{ for } \varepsilon_{el,vol} < 0$$

where $\varepsilon_{el,vol}$ is the volumetric strain, K_c is the bulk modulus for compression, and K_t the bulk modulus for tension.

UNIAXIAL DATA

Many nonlinear stress–strain curves are measured in a tensile test, for which a nonlinear curve of force vs displacement is obtained.

If only the uniaxial behavior is measured, the measurements do not fully define the material behavior. An extra assumption is needed. The *Uniaxial data* material model allows you to assume either a constant Poisson's ratio, or a constant bulk modulus. Also, if only uniaxial extension data is available, further assumptions are needed for covering the uniaxial compressive behavior of the material.

For the uniaxial tensile test, the axial stress corresponds to the principal stresses $\sigma_{ax} = \sigma_1 = \sigma_{mises}$, and the other two principal stresses are equal to zero, $\sigma_2 = \sigma_3 = 0$.

The principal (axial) strain is positive in tension, $\varepsilon_{ax} = \varepsilon_1$, and the other two (transverse) strains are negative and related by the Poisson's ratio $\varepsilon_2 = \varepsilon_3 = -\nu\varepsilon_1$.

For uniaxial compression, the axial strain is negative, and when the principal strains are sorted as $\varepsilon_1 > \varepsilon_2 > \varepsilon_3$ it corresponds to the third principal strain, $\varepsilon_{ax} = \varepsilon_3$. The other two (transverse) strains are positive and related by the Poisson's ratio $\varepsilon_1 = \varepsilon_2 = -\nu\varepsilon_3$. Also, the axial stress is negative in compression, and it corresponds to the third principal stress $\sigma_{ax} = \sigma_3 = -\sigma_{mises}$. The other two transverse stresses are zero $\sigma_1 = \sigma_2 = 0$.

Other strain measures that can be obtained from the elastic strain tensor or its principal values, are the elastic volumetric strain

$$\varepsilon_{vol,el} = \text{trace}(\varepsilon_{el})$$

and the elastic shear strain,

$$\gamma_{el} = 2\sqrt{J_2(\varepsilon_{el})} = \sqrt{2\text{dev}(\varepsilon_{el}):\text{dev}(\varepsilon_{el})}\,.$$

These are used to define the elastic axial strain variable for multiaxial loading.

The uniaxial test defines the relation between the axial stress and elastic axial strain as

$$\sigma_{ax} = E_s \varepsilon_{ax}$$

Here, E_s is the secant Young's modulus, and the axial stress σ_{ax} is considered as a function of the elastic axial strain ε_{ax}. Thus

$$E_s = \frac{\sigma_{ax}}{\varepsilon_{ax}}$$

At zero strain, the secant Young's modulus is defined as

$$E_s = \left.\frac{d\sigma_{ax}}{d\varepsilon_{ax}}\right|_{\varepsilon_{ax} = 0}$$

Assuming a constant Poisson's ratio, the secant shear modulus is defined as

$$G_s = \frac{E_s}{2(1+\nu)}$$

and the secant bulk modulus as

$$K_s = \frac{E_s}{3(1-2\nu)}$$

Furthermore, if only tensile stress–strain data is available, the elastic axial strain for multiaxial loading is computed from the elastic shear strain γ and Poisson's ratio ν as

$$\varepsilon_{ax} = \frac{\sqrt{3}\gamma}{2(1+\nu)}$$

When nonsymmetric stress–strain data is available, the elastic axial strain for multiaxial loading is computed from the elastic volumetric strain and Poisson's ratio

$$\varepsilon_{ax} = \frac{\varepsilon_{vol}}{1-2\nu}$$

as this expression captures the change of sign in the elastic axial strain when changing from a tensile to a compressive state.

When using a constant bulk modulus assumption, only the symmetric part from the stress–strain data is considered. The secant shear modulus is instead defined as

$$G_s = \frac{3KE_s}{9K - E_s}$$

and the elastic axial strain is defined from both volumetric and shear elastic strains

$$\varepsilon_{ax} = \frac{\gamma}{\sqrt{3}} + \frac{|\varepsilon_{vol}|}{3}$$

It is possible to use any uniaxial data function to define the axial stress as a function of elastic axial strain

$$\sigma_{ax} = \sigma_{ax}(\varepsilon_{ax})$$

provided that

$$\sigma_{ax} = 0 \text{ when } \varepsilon_{ax} = 0$$

The elastic axial strain ε_{ax} can be called in user defined uniaxial stress functions by referencing the variables `solid.eax`, where `solid` is the name of the physics interface node. See also the description of the Uniaxial Data material model in the Solid Mechanics interface documentation.

SHEAR DATA

Many nonlinear stress–strain curves are measured in a shear test, for which a nonlinear curve of force vs displacement is obtained.

If only the shear behavior is measured, the measurements do not fully define the material behavior and therefore an extra assumption is needed. The *Shear data* material model assumes a constant bulk modulus.

Other strain measures that can be obtained from the strain tensor or its principal values, are the elastic volumetric strain

$$\varepsilon_{vol,el} = \text{trace}(\varepsilon_{el})$$

and the elastic shear strain,

$$\gamma = 2\sqrt{J_2(\varepsilon_{el})} = \sqrt{2\text{dev}(\varepsilon_{el}):\text{dev}(\varepsilon_{el})}\,.$$

These are used to define the nonlinear stress–strain relation for multiaxial loading.

From the shear test one could define the relation between the shear stress τ and the elastic shear strain γ as

$$\tau = G_s\gamma$$

Here, the secant shear modulus G_s is constant for Linear Elastic materials, but in general one could use a nonlinear relation. It is possible to use shear data to define the shear stress as a function of elastic shear strain as follows:

$$\tau = \tau(\gamma)$$

provided that

$$\tau = 0 \text{ when } \gamma = 0$$

in which case, the secant shear modulus is computed from

$$G_s = \frac{\tau(\gamma)}{\gamma}$$

With the help of the secant shear modulus G_s computed from shear data, Hooke's law simplifies to

$$\sigma = 2G_s\text{dev}(\varepsilon_{el}) + K\text{trace}(\varepsilon_{el})I$$

where K is the bulk modulus.

	The elastic shear strain γ can be called in user defined shear stress functions by referencing the variables `solid.esh`, where `solid` is the name of the physics interface node. See also the description of the Shear Data material model in the Solid Mechanics interface documentation.

HYPERBOLIC LAW

A hyperbolic relation between shear stress and shear strain is obtained by setting the secant shear modulus

$$G_s = G\frac{1}{1+\left(\frac{\gamma}{\gamma_{ref}}\right)^n}$$

where the strain exponent n and a reference shear strain γ_{ref} control the shape of the hyperbola.

For hyperbolic material models, the maximum shear modulus occurs at zero shear strain, so practitioners might call G the "maximum shear modulus" and use the notation $G_{\max}$. Sometimes it is also called "small-strain shear modulus".

HARDIN–DRNEVICH

The Hardin–Drnevich model (Ref. 87) is a hyperbolic soil model (with $n = 1$) defined by two input parameters: the initial shear modulus G and a reference shear strain γ_{ref}:

$$G_{\text{s}} = G\frac{1}{1+\frac{\gamma}{\gamma_{\text{ref}}}}$$

This nonlinear soil model is commonly used for modeling soil dynamics in earthquake engineering problems.

Since $\tau = G_{\text{s}}\gamma$, the shear stress is bounded by $\tau_{\max} = G\gamma_{\text{ref}}$ as the shear strain increases.

The hyperbolic Hardin–Drnevich model is normally used for quantifying stiffness reduction curves in soils. Commonly, the reference shear strain γ_{ref} is replaced by the reference shear strain at which the secant shear modulus has been decreased to 70% of its initial value. Calling this shear strain value $\gamma_{0.7}$, the reference strain is written as

$$\gamma_{\text{ref}} = \frac{7}{3}\gamma_{0,7}$$

and the secant shear modulus as

$$G_{\text{s}} = G\frac{1}{1+\frac{3}{7}\frac{\gamma}{\gamma_{0,7}}}$$

so that when $\gamma = \gamma_{0.7}$ the secant shear modulus is $G_{\text{s}} = 0.7G$.

DUNCAN–CHANG

The original model was originated by Kondner to fit triaxial test data for undrained soils. Duncan and Chang (Ref. 88) and other coworkers (Ref. 89) developed this hyperbolic model to its current state. The material model is written in terms of the axial and radial stresses σ_1 and σ_3 and the axial strain ε, and it describes the stress–strain curve by fitting the hyperbola

$$\sigma_1 - \sigma_3 = \frac{\varepsilon}{a + b\varepsilon}$$

here a and b are material parameters obtained by curve fitting data from the triaxial test. The parameter a is related to the initial Young's modulus E

$$a = \frac{1}{E}$$

and the parameter b defines the asymptote of the hyperbola, which is related to the ultimate value of $\sigma_1 - \sigma_3$ denoted q_{ult}

$$\frac{1}{b} = (\sigma_1 - \sigma_3)_{\text{ult}} = q_{\text{ult}}$$

The ultimate value q_{ult} is related to the strength of the soil.

For the triaxial test, the axial strain ε is related to the shear strain γ by the Poisson's ratio as

$$\varepsilon = \frac{\sqrt{3}}{2(1+\nu)}\gamma$$

and the axial and radial stresses are related to the shear stress as $\sigma_1 - \sigma_3 = \sqrt{3}\tau$.

It is possible then to write the relation between shear stress and shear strain as

$$\tau = \frac{\dfrac{1}{2(1+\nu)}\gamma}{\dfrac{1}{E} + \dfrac{1}{q_{\text{ult}}}\dfrac{\sqrt{3}}{2(1+\nu)}\gamma}$$

Since the initial shear modulus is related to the initial Young's modulus as $G = E/2(1+\nu)$, this stress–strain relation can alternatively be written as

$$\tau = \frac{\gamma}{\dfrac{1}{G} + \dfrac{\sqrt{3}\gamma}{q_{\text{ult}}}}$$

which is an hyperbolic law with a secant shear modulus of

$$G_s = \frac{G}{1 + \dfrac{G}{q_{\text{ult}}}\sqrt{3}\gamma}$$

DUNCAN–SELIG

The Duncan–Selig model is a combination of the Duncan's hyperbolic material model (Ref. 88, Ref. 89) and Selig's model to describe nonlinear bulk modulus behavior. Selig (Ref. 90) further developed the model of Duncan and others in order to include a nonlinear volumetric response in soils.

The model defines the nonlinear volumetric response for the pressure as

$$p = -K\frac{\varepsilon_{el,vol}}{1-\frac{\varepsilon_{el,vol}}{\varepsilon_{ult}}}$$

where $\varepsilon_{el,vol}$ is the volumetric strain, and ε_{ult} is the asymptote of the hyperbola, the maximum value for the volumetric compression. Note that K represents the bulk modulus at zero strain.

The secant (nonlinear) bulk modulus is defined for this material model as

$$K_s = K\frac{1}{1-\frac{\varepsilon_{el,vol}}{\varepsilon_{ult}}}$$

SMALL-STRAIN OVERLAY

The majority of soils exhibit a nonlinear stiffness behavior at small strains, coupled with a hysteresis effect. This phenomenon is characterized by an exceptionally high shear modulus at low strains, commonly referred to as *small-strain stiffness*. This small-strain stiffness nonlinearly decays as the strain level increases.

The small-strain overlay model (Ref. 79, Ref. 80) incorporates both the nonlinear stiffness and hysteresis effects by utilizing a nonlinear shear modulus derived from the shear strain history observed over multiple loading–unloading cycles.

The stress–strain relationship is defined by

$$\sigma = 2G\,\mathrm{dev}(\varepsilon_{el,dev} - \varepsilon_{r,dev}) + K(\varepsilon_{el,vol} - \varepsilon_{r,vol})I + \sigma_r$$

where $\varepsilon_{el,dev}$ and $\varepsilon_{el,vol}$ are the deviatoric and volumetric elastic strains, and $\varepsilon_{r,dev}$, $\varepsilon_{r,vol}$, and σ_r are the deviatoric and volumetric elastic strains and stress at the last load reversal point.

The secant shear modulus is defined from an Hyperbolic Law, in a similar way as done for the Hardin–Drnevich model. The *secant shear modulus for primary loading* is

defined by two input parameters, the initial shear modulus G_0 and a reference shear strain γ_{ref}, as

$$G_{pl} = G_0 \frac{1}{1 + \frac{\gamma_{hist}}{\gamma_{ref}}}$$

Here, γ_{hist} is a measure of the shear strain history. In Ref. 79, the reference shear strain is defined from $\gamma_{ref} = \gamma_{0.7}/\alpha$, where $\alpha = 0.385$ is a tuning parameter that fits experimental data and $\gamma_{0.7}$ is the shear strain at which the secant shear modulus has been decreased to 70% of its initial value.

The *secant shear modulus for unloading and reloading* is defined as

$$G_{ur} = G_0 \frac{1}{1 + \frac{\gamma_{hist}}{2\gamma_{ref}}}$$

which is higher than G_{pl}.

The load path is considered as *primary loading* if the soil has not reached any load reversal stage, in which case G_{pl} is used. When a load reversal happens, the soil is assumed to be on the unloading–reloading path, and G_{ur}, is instead used.

USER DEFINED

This option allows you to write explicitly how the pressure depends on the elastic volumetric strain. This could be an analytic function or data interpolated from a table.

The elastic volumetric strain $\varepsilon_{vol,el}$ can be called in user defined expressions by referencing the variables `solid.eelvol`, where `solid` is the name of the physics interface node.

Linear Viscoelasticity

Viscoelastic materials have a time-dependent response even if the loading is constant in time. Many polymers and biological tissues exhibit this behavior. *Linear viscoelasticity* is a commonly used approximation, where the stress depends linearly on the strain and its time derivatives (strain rate). Also, linear viscoelasticity deals with the *additive decomposition* of stresses and strains. It is usually assumed that the viscous

part of the deformation is incompressible so that the volumetric deformation is purely elastic, but it is also possible to account for volumetric viscoelasticity.

For isotropic linear elastic materials in the absence of inelastic stresses, Hooke's law in Equation 3-21 reduces to

$$\sigma = C : \varepsilon_{el}$$

where the elastic strain tensor $\varepsilon_{el} = \varepsilon - \varepsilon_{inel}$ represents the difference between the total strain and inelastic strains, such as thermal strains.

The stress tensor can be decomposed into a pressure and a deviatoric stress:

$$\sigma = -pI + \sigma_{dev}$$

The pressure (mean stress or volumetric stress) is given with a positive sign in compression

$$p = -\frac{1}{3}\text{trace}(\sigma)$$

and the deviatoric stress is computed from the total stress minus the volumetric contribution

$$\sigma_{dev} = \text{dev}(\sigma) = \sigma - \frac{1}{3}\text{trace}(\sigma) = \sigma + pI$$

The elastic strain tensor ε_{el} can in the same way be decomposed into volumetric and deviatoric components

$$\varepsilon_{el} = \frac{1}{3}\varepsilon_{el,vol}I + \varepsilon_{el,dev}$$

with the volumetric elastic strain defined as

$$\varepsilon_{el,vol} = \text{trace}(\varepsilon_{el})$$

and the deviatoric contribution

$$\varepsilon_{el,dev} = \text{dev}(\varepsilon_{el})$$

For isotropic linear elastic materials, the pressure is then related to the volumetric elastic deformation by the bulk modulus K

$$p = -\frac{1}{3}\text{trace}(\sigma) = -K\text{trace}(\varepsilon_{el}) = -K\varepsilon_{el,vol}$$

and the deviatoric stress tensor is linearly related to the deviatoric elastic strain tensor by the shear modulus G

$$\sigma_{dev} = \text{dev}(\sigma) = 2G\text{dev}(\varepsilon_{el}) = 2G\varepsilon_{el,dev}$$

The total stress in Equation 3-21 is then

$$\sigma = K\varepsilon_{el,vol}I + 2G\varepsilon_{el,dev}$$

In case of geometric nonlinearity, σ represents the second Piola–Kirchhoff stress tensor and ε_{el} the elastic Green–Lagrange strain tensor.

THE GENERALIZED MAXWELL MODEL

For deviatoric viscoelastic materials, the deviatoric stress σ_d is not linearly related to the deviatoric strain ε_d but it also depends on the strain history. It is normally defined by the hereditary integral:

$$\sigma_d = 2\int_0^t \Gamma(t-t')\frac{\partial \varepsilon_d}{\partial t'}dt'$$

The function $\Gamma(t)$ is called the *relaxation shear modulus function* (or just *relaxation function*) and it can be found by measuring the stress evolution in time when the material is held at a constant strain.

The relaxation function is often approximated by a Prony series:

$$\Gamma(t) = G + \sum_{m=1}^{N} G_m \exp\left(-\frac{t}{\tau_m}\right)$$

A physical interpretation of this approach, often called the *generalized Maxwell model*, is shown in Figure 3-1

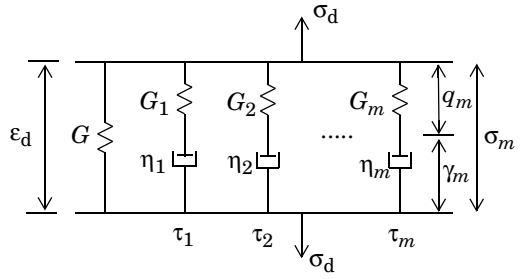


Figure 3-1: Generalized Maxwell model.

Hence, G is the stiffness of the main elastic branch, G_m represents the stiffness of the spring in branch m, and τ_m is the *relaxation time* constant of the spring-dashpot pair in branch m.

The relaxation times τ_m are normally measured in the frequency domain, so the viscosity of the dashpot is not a physical quantity but instead it is derived from stiffness and relaxation time measurements. The viscosity of each branch can be expressed in terms of the shear modulus and relaxation time as

$$\eta_m = G_m \tau_m$$

The auxiliary strain variable q_m is introduced to represent the extension of the corresponding abstract spring, and the auxiliary variables $\gamma_m = \varepsilon_d - q_m$ represent the extensions in the dashpots.

The shear modulus of the elastic branch G is normally called the *long-term shear modulus*, or *steady-state stiffness*, and it is often denoted with the symbol G_∞. The *instantaneous* shear modulus G_0 is then defined as the long-term shear modulus plus the sum of the stiffnesses of all the viscoelastic branches

$$G_0 = G + \sum_{m=1}^{N} G_m$$

This is the equivalent stiffness when an external load is applied much faster than the shortest relaxation time of any viscoelastic branch.

Sometimes, the relaxation function $\Gamma(t)$ is expressed in terms of the instantaneous stiffness and relative weights, so that the Prony series is given as

$$\Gamma(t) = G_0\left(w_\infty + \sum_{m=1}^{N} w_m \exp\left(-\frac{t}{\tau_m}\right)\right)$$

In this case, the long-term shear modulus is related to the instantaneous shear modulus by the weight $w_\infty < 1$

$$G = w_\infty G_0$$

and the shear moduli in each branch are defined by the weights w_m

$$G_m = w_m G_0$$

It must be assumed then that the weights fulfill the constraint

$$w_\infty + \sum_{m=1}^{N} w_m = 1$$

The long-term shear modulus G is given in the parent material model (for example, Linear Elastic Material). If your material data consists of the instantaneous shear modulus G_0 and the weights w_m, you can convert the data using the formulas above.

If your data consists of Young's modulus and relaxation time per branch, you can convert it to the equivalent shear moduli. As the strain tensors in the dashpots are isochoric (volume preserving) it means that $G_m = E_m/3$.

In the parent Linear Elastic Material, you can enter the long-term elastic data in any form, for example in terms of Young's modulus, E, and Poisson's ratio, ν. It will then be converted to the corresponding shear modulus G internally.

The stress per branch can be written either in terms of the strain in the spring, q_m, or the strain in the dashpot, γ_m

$$\sigma_m = 2G_m q_m = 2\eta_m \dot{\gamma}_m = 2G_m(\varepsilon_d - \gamma_m)$$

The sum of the stresses in the viscoelastic branches is then computed from

$$\sigma_q = \sum_{m=1}^{N} \sigma_m = \sum_{m=1}^{N} 2G_m(\varepsilon_d - \gamma_m)$$

The total stress in Hooke's law (Equation 3-21) is then augmented by the viscoelastic stress σ_q

$$\sigma = \sigma_0 + C : \varepsilon_{el} + \sigma_q \tag{3-32}$$

The same derivation is valid for a volumetric viscoelastic response; see Volumetric Response.

In the case of mixed volumetric and deviatoric viscoelasticity, it is assumed that the relaxation times per branch are the same for the deviatoric and volumetric deformation.

Computing the Stress in Each Branch

For deviatoric viscoelasticity, the auxiliary variable γ_m is a symmetric isochoric (volume preserving) strain tensor, which has as many components as the number of strain components of the problem class. Since the stress per branch is written as

$$\sigma_m = 2G_m q_m = 2\eta_m \dot{\gamma}_m = 2G_m(\varepsilon_d - \gamma_m)$$

the auxiliary variables γ_m can be computed by solving the ODE

$$\eta_m \dot{\gamma}_m = G_m(\varepsilon_d - \gamma_m) \tag{3-33}$$

The relation between viscosity and relaxation time is

$$\eta_m = G_m \tau_m$$

so that Equation 3-33 can equivalently be written as

$$\tau_m \dot{\gamma}_m + \gamma_m = \varepsilon_d \tag{3-34}$$

The viscoelastic strain variables γ_m are called for instance `solid.lemm1.vis1.ev1_11`, where `solid` is the tag of the physics interface, `lemm1` is the tag of the linear elastic material, and `vis1` is the tag of the viscoelasticity node. The trailing numbers indicate the branch number and the strain tensor indices.

Volumetric Response

It is usually assumed that the viscous part of the deformation is incompressible so that the volumetric deformation is purely elastic, but this does not have to be the case, and the same derivation as stated in previous sections can be applied to the volumetric response.

For volumetric viscoelasticity, the bulk modulus of the elastic branch K is normally called the *long-term bulk modulus*, or *steady-state stiffness*, and it is often denoted with the symbol K_∞. The *instantaneous* bulk modulus K_0 is then defined as the long-term bulk modulus plus the sum of the bulk moduli of all the viscoelastic branches.

The scalar equation for the volumetric deformation per branch, e_m, is

$$\tau_m \dot{e}_m + e_m = \varepsilon_{\text{el,vol}} \tag{3-35}$$

where τ_m is the relaxation time per branch and $\varepsilon_{\text{el,vol}}$ is the elastic volumetric strain.

The sum of the stresses in the viscoelastic branches is then computed from

$$\sigma_q = \sum_{m=1}^{N} \sigma_m = \sum_{m=1}^{N} K_m(\varepsilon_{\text{el,vol}} - e_m)I$$

where K_m is the bulk modulus of branch m.

The volumetric viscoelastic strain variables e_m are called, for example, `solid.lemm1.vis1.evvol1`, where `solid` is the tag of the physics interface, `lemm1` is the tag of the linear elastic material node, and `vis1` is the tag of the viscoelasticity node. The trailing number represents the branch number (that is, the first branch in this case).

Equation 3-34 and Equation 3-35 are solved using a Local Time Integration algorithm. Alternatively, the viscoelastic strain variables, γ_m and e_m, can be treated as additional degrees of freedom (DOFs), in which case is it possible to select *Gauss-point data* or *Discontinuous Lagrange* shape functions (see Shape Functions and Element Types for details).

Using a Suitable Number of Branches

Only branches that have a relaxation time which is of the same order of magnitude as the temporal variation of the loading are important for the viscoelastic response. The

cost in terms of memory consumption and computational speed increase significantly if the model contains many branches, particularly in time domain and for eigenfrequency analyses. In these cases many auxiliary equations of the form Equation 3-34 have to be solved for.

However, if a viscoelastic branch relaxes much faster or much slower than the time scale of the excitation period of external loads, the corresponding dashpot may be considered to be either fully relaxed or rigid, depending on the relaxation time of the branch relative to the excitation frequency. These fully relaxed or rigid branches can then be removed and grouped into equivalent branches, thus reducing the computational cost.

When the frequency band of interest is bounded by two cutoff frequencies, f_{lower} and f_{upper}, it is possible to prune viscoelastic branches that have relaxation times such as

$$\frac{1}{f_{\text{high}}} < \tau_m < \frac{1}{f_{\text{low}}}$$

Branches with short relaxation times τ_k represent the response of the viscoelastic material to high frequency excitations. When the external loads represent excitations bounded by an upper frequency f_{upper}, it is possible to prune branches with relaxation times that fulfill the relation $\tau_k f_{\text{upper}} < 1$. These branches are grouped into an equivalent branch, with relaxation time and stiffness such as

$$\tau_{\text{high}} = \frac{\sum_k G_k \tau_k^2}{\sum_k G_k \tau_k} \text{ and } G_{\text{high}} = \frac{1}{\tau_{\text{high}}} \sum_k G_k \tau_k$$

Viscoelastic branches with relaxation times τ_k such that $1 < \tau_k f_{\text{low}}$ are grouped into an equivalent branch forming the low frequency cutoff. The stiffness and relaxation time for the equivalent branch are computed from

$$\frac{1}{\tau_{\text{low}}} = \frac{1}{G_{\text{low}}} \sum_k \frac{G_k}{\tau_k} \text{ and } G_{\text{low}} = \sum_k G_k$$

Frequency Domain

In the frequency domain, the relaxation function becomes complex valued. The complex shear modulus for the generalized Maxwell model is then defined as the sum of the shear modulus in the pure elastic branch plus the complex shear moduli in the viscoelastic branches

$$G_{\mathrm{GM}} = G + \sum_{m=1}^{N} G_m \frac{j\omega\tau_m}{1 + j\omega\tau_m}$$

The storage and loss moduli are then computed as the real and imaginary parts of the complex shear modulus

$$G' = \mathrm{real}(G_{\mathrm{GM}}) = G + \sum_{m=1}^{N} G_m \frac{(\omega\tau_m)^2}{1 + (\omega\tau_m)^2}$$

and

$$G'' = \mathrm{imag}(G_{\mathrm{GM}}) = \sum_{m=1}^{N} G_m \frac{\omega\tau_m}{1 + (\omega\tau_m)^2}$$

Energy Dissipation

The *dissipated energy density rate* (SI unit: $\mathrm{W/m^3}$) in each dashpot m is

$$\dot{W}_m = \sigma_m : \dot{\gamma}_m$$

The rate of total dissipated energy density in the Generalized Maxwell material is then

$$\dot{W}_{\mathrm{v}} = \sum_{m=1}^{N} \dot{W}_m$$

In order to compute the dissipated energy density, the variable $\dot{W}_{\mathrm{v}}$ is integrated over time. For frequency domain studies, the dissipation of viscous forces averaged over a time period $2\pi/\omega$ is computed from the shear loss modulus G'' as

$$W_{\mathrm{v}} = \omega G'' \tilde{\varepsilon}_{\mathrm{d}} : \mathrm{conj}(\tilde{\varepsilon}_{\mathrm{d}})$$

THE MAXWELL MODEL

The *Maxwell* model (or *Maxwell* element) consists of a spring and a dashpot arranged in series. Maxwell elements are the building blocks used in The Generalized Maxwell Model.

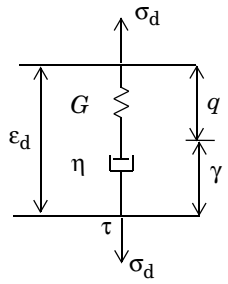


Figure 3-2: Maxwell model.

In frequency domain, the compliance of the Maxwell model is defined as the sum of the compliance modulus of the elastic part plus the compliance in the damper

$$J_{\mathrm{M}} = \frac{1}{G} + \frac{1}{j\omega\tau G}$$

The complex shear modulus then reads

$$G_{\mathrm{M}} = \frac{1}{J_{\mathrm{M}}} = \frac{j\omega\tau}{1 + j\omega\tau} G$$

The storage and loss moduli are defined as the real and imaginary parts of the complex shear modulus

$$G' = \mathrm{real}(G_{\mathrm{M}}) = \frac{(\omega\tau)^2}{1 + (\omega\tau)^2} G$$

and

$$G'' = \mathrm{imag}(G_{\mathrm{M}}) = \frac{\omega\tau}{1 + (\omega\tau)^2} G$$

THE GENERALIZED KELVIN–VOIGT MODEL

The *generalized Kelvin–Voigt* model is used to simulate the viscoelastic deformation in a wide range of materials such as concrete, biological tissues, and glassy polymers.

Just as for the generalized Maxwell model, the deviatoric strain ε_d is not linearly related to the deviatoric stress σ_d, but it also depends on the strain history. It is normally defined by the hereditary integral:

$$\varepsilon_d = \int_0^t \psi(t - t') \frac{\partial \sigma_d}{2 \partial t'} dt'$$

The function $\psi(t)$ is called the *compliance function* (also called *creep compliance* or *creep function*) and it can be found by measuring the strain evolution in time when the material is held at a constant stress.

The compliance function is often approximated by a Prony series:

$$\psi(t) = J + \sum_{m=1}^{N} J_m \left(1 - \exp\left(-\frac{t}{\tau_m}\right)\right)$$

A physical interpretation of this rheological model consists of an elastic branch plus a number of Kelvin–Voigt elements arranged in series, this approach is shown in Figure 3-3.

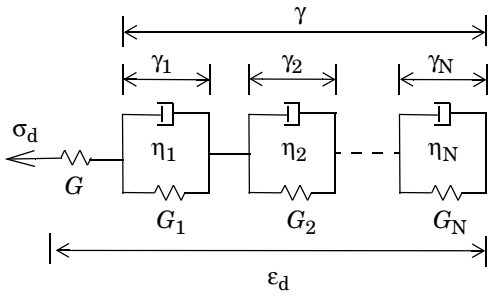


Figure 3-3: The generalized Kelvin–Voigt model.

Here, G is the stiffness of the main elastic branch, G_m represents the stiffness of the spring in element m, and τ_m is the *relaxation time* constant of the spring-dashpot pair in branch m. The compliance J in the pure elastic branch is related to the shear modulus by

$$J = \frac{1}{G}$$

and the compliances per branch are related to the shear moduli as

$$J_m = \frac{1}{G_m}$$

The relaxations time τ_m is normally measured in the frequency domain, so the viscosity of the dashpot is not a physical quantity but instead it is derived from stiffness and relaxation time measurements. The viscosity in each branch can be expressed in terms of the stiffness or compliance modulus and relaxation time as

$$\eta_m = G_m \tau_m = \tau_m / J_m$$

The auxiliary strain variables γ_m are introduced to represent the extension of the corresponding Kelvin–Voigt element. Since the elements are arranged in series, the total viscoelastic strain is given as the sum of the auxiliary strains

$$\gamma = \sum_m \gamma_m \tag{3-36}$$

The deviatoric stress σ_d in the pure elastic branch is the same as in all the Kelvin–Voigt elements

$$\sigma_d = 2G\,\mathrm{dev}(\varepsilon_{el}) = 2G_m \gamma_m + 2\eta_m \dot{\gamma}_m \tag{3-37}$$

Here, the deviatoric elastic strain is defined by the difference between the total and inelastic strains.

Computing the Strain in Each Branch

The auxiliary strain variables γ_m represent a symmetric isochoric (volume preserving) strain tensor, which has as many components as the number of strain components of the problem class.

The stress in each element m is given by the sum of the stresses in the spring and dashpot arranged in parallel

$$\sigma_d = 2G_m \gamma_m + 2\eta_m \dot{\gamma}_m = 2G_m(\gamma_m + \tau_m \dot{\gamma}_m)$$

Here, γ_m is the strain in the element m, and $\eta_m = \tau_m G_m$ is the viscosity. The auxiliary variables γ_m are computed by solving the ODE

$$\gamma_m + \tau_m \dot{\gamma}_m = \frac{1}{2G_m}\sigma_d \tag{3-38}$$

The compliance of the elastic branch, $J = 1/G$, is normally called the *instantaneous compliance*, and it is often denoted with the symbol J_0. This gives the equivalent stiffness when the material is loaded by an abrupt load much faster than the shortest relaxation time of any branch.

The long-term compliance J_∞ is defined as the instantaneous compliance plus the sum of the compliances of the viscoelastic branches

$$J_\infty = J + \sum_{m=1}^{N} J_m = \frac{1}{G} + \sum_{m=1}^{N} \frac{1}{G_m}$$

The long-term shear modulus then reads $G_\infty = 1/J_\infty$.

Sometimes, the compliance function $\psi(t)$ is expressed in terms of the instantaneous compliance J and relative weights w_m, so that the Prony series reads

$$\psi(t) = J\left(1 + \sum_{m=1}^{N} w_m\left(1 - \exp\left(-\frac{t}{\tau_m}\right)\right)\right)$$

In this case, the compliance and shear modulus in each branch are

$$J_m = w_m J \text{ and } G_m = G/w_m$$

and the long-term shear compliance and long-term shear modulus read

$$J_\infty = J\left(1 + \sum_{m=1}^{N} w_m\right) \text{ and } G_\infty = G/\left(1 + \sum_{m=1}^{N} w_m\right)$$

Frequency Domain

In frequency domain, the compliance function becomes complex valued. The complex compliance for the generalized Kelvin–Voigt model is then defined as the sum of the compliance in the pure elastic element (parent material) plus the complex compliances in the viscoelastic elements.

$$J_{\text{GVK}} = \frac{1}{G} + \sum_{m=1}^{N} \frac{1}{G_m} \frac{1}{1 + j\omega\tau_m}$$

The storage and loss compliances are then computed as the real and imaginary parts of the complex compliance function

$$J' = \text{real}(J_{\text{GVK}}) = \frac{1}{G} + \sum_{m=1}^{N} \frac{1}{G_m} \frac{1}{1 + (\omega\tau_m)^2} = \frac{1}{G}\left(1 + \sum_{m=1}^{N} \frac{w_m}{1 + (\omega\tau_m)^2}\right)$$

and

$$J'' = \text{imag}(J_{\text{GVK}}) = \sum_{m=1}^{N} \frac{1}{G_m} \frac{-\omega\tau_m}{1 + (\omega\tau_m)^2} = \frac{1}{G} \sum_{m=1}^{N} \frac{-w_m \omega\tau_m}{1 + (\omega\tau_m)^2}$$

Energy Dissipation

The *dissipated energy density rate* (SI unit: W/m^3) in each branch m is given by the dissipation in the dashpot

$$\dot{W}_m = 2\eta_m \dot{\gamma}_m : \dot{\gamma}_m$$

The rate of total dissipated energy density in the generalized Kelvin–Voigt material is then

$$\dot{W}_{\text{v}} = \sum_{m=1}^{N} \dot{W}_m$$

In order to compute the dissipated energy density, the variable $\dot{W}_{\text{v}}$ is integrated over time. For frequency domain studies, the dissipation of viscous forces averaged over a time period $2\pi/\omega$ is computed from the shear loss modulus G'' as

$$W_{\text{v}} = \omega G'' \tilde{\varepsilon}_{\text{d}} : \text{conj}(\tilde{\varepsilon}_{\text{d}})$$

THE KELVIN–VOIGT MODEL

The *Kelvin–Voigt viscoelastic model* is represented by a spring connected in parallel with a dashpot:

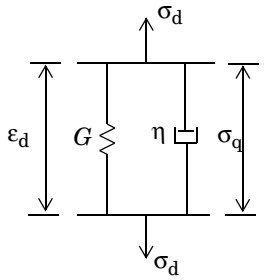


Figure 3-4: The Kelvin–Voigt model.

The stress tensor in the viscous branch is computed from the elastic strain rate

$$\sigma_q = 2\eta\dot{\varepsilon}_d = 2G\tau\dot{\varepsilon}_d \tag{3-39}$$

so there is no need to add the extra DOFs to compute the auxiliary strain tensor γ.

The relaxation time relates the viscosity and shear modulus by $\eta = \tau G$. The equivalent shear modulus is used in case of an anisotropic linear elastic material.

The dissipated energy density rate of the Kelvin–Voigt model is then computed from its rate

$$\dot{W}_v = \sigma_q : \dot{\varepsilon}_d$$

The Kelvin–Voigt model is equivalent to viscous damping; see Viscous Damping.

STANDARD LINEAR SOLID MODEL

The *standard linear solid model*, also called *SLS model*, *Zener model*, or *three-parameter model*, is a simplification of the generalized Maxwell model with only one spring-dashpot branch:

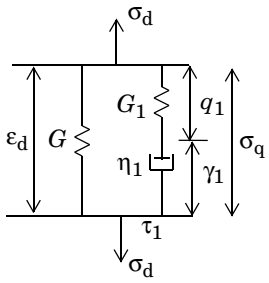


Figure 3-5: Standard linear solid (SLS) model.

The stress in the viscoelastic branch is computed as

$$\sigma_q = 2G_1 q_1 = 2\eta_1 \dot{\gamma}_1 = 2G_1(\varepsilon_d - \gamma_1)$$

where the relaxation time is related to the stiffness and relaxation time as $\eta_1 = \tau_1 G_1$.

The auxiliary strain tensor γ_1 is computed by solving by the ODE

$$\tau_1 \dot{\gamma}_1 + \gamma_1 = \varepsilon_d$$

and the dissipated energy density rate of the single branch is calculated from

$$\dot{W}_v = \sigma_q : \dot{\gamma}_1$$

The long-term shear modulus $G_\infty = G$ is given in the parent Linear Elastic Material, and the instantaneous stiffness is given by the sum $G_0 = G + G_1$.

The Standard linear solid model is equivalent to damping; see Maximum Loss Factor.

THE BURGERS MODEL

The *Burgers model* consists of a Maxwell (spring-dashpot) branch in series with a Kelvin–Voigt branch. The rheological representation of this material model is shown in Figure 3-6:

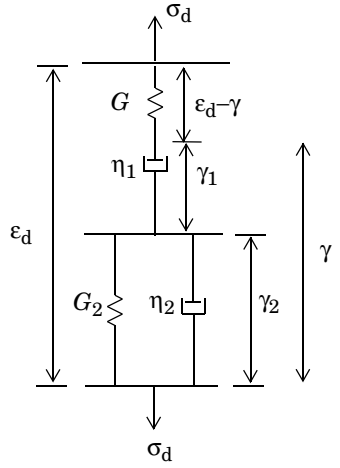


Figure 3-6: The Burgers model.

The strain in the first dashpot follows the ODE

$$\mathrm{dev}(\sigma) = 2G(\varepsilon_d - \gamma) = 2\eta_1\dot{\gamma}_1$$

where the shear modulus G is taken from the parent Linear Elastic material.

The strain in the second dashpot follows the ODE

$$2\eta_2\dot{\gamma}_2 = \mathrm{dev}(\sigma) - 2G_2\gamma_2$$

The total strain in the dashpots is computed from

$$\gamma = \gamma_1 + \gamma_2$$

Combining these equations, it is possible to recover a second-order ODE for the strain tensor γ:

$$\ddot{\gamma}_2 = \left(\frac{1}{2\eta_1} + \frac{1}{2\eta_2}\right)\mathrm{dev}(\dot{\sigma}) - \frac{G_2}{\eta_2}\dot{\gamma} + \frac{G_2}{2\eta_1\eta_2}\mathrm{dev}(\sigma)$$

Note that the Burgers material has two relaxation times related to the stiffness and viscosity in the springs and dashpots. The relaxation times τ_1 and τ_2 are related to the stiffness and viscosities as $\tau_1 = \eta_1 / G$ and $\tau_2 = \eta_2 / G_2$.

The instantaneous stiffness $G_0 = G$ is given in the parent Linear Elastic Material.

GENERALIZED MAXWELL MODEL WITH FRACTIONAL DERIVATIVES

Typically, the rheology of viscoelasticity models consists of springs and dashpots arranged in series and in parallel. Using the framework of *fractional calculus*, the constitutive equations of linear viscoelasticity can be generalized with a new type of element, named *spring-pot* (Ref. 50). In some cases, viscoelasticity models with fractional derivatives have shown to better match experimental data.

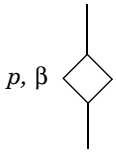


Figure 3-7: Spring-pot element.

The basic stress–strain relation of the spring-pot element is given by a proportionality factor p and a *fractional time derivative* of order β

$$\sigma = p \frac{d^\beta}{dt^\beta} \varepsilon = p D^\beta \varepsilon$$

The fractional order β takes a value between $0 < \beta < 1$. For $\beta = 0$, the parameter p plays the role of a stiffness in a spring, and for $\beta = 1$ the parameter plays the role of the viscosity in a dashpot. The SI unit for such material parameter would be $\text{Pa}\cdot\text{s}^\beta$.

For a pair of one spring and one spring-pot element connected in series, a so-called Maxwell element with a fractional time derivative, the deviatoric stress strain relation is given by

$$\text{dev}(\sigma) = 2 p D^\beta \gamma = 2G(\varepsilon_d - \gamma)$$

The rheological representation of this arrangement is shown in Figure 3-8

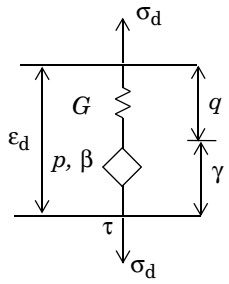


Figure 3-8: A spring-pot element connected in series with a spring.

In frequency domain, the fractional time derivative operator is substituted by

$$D^{\beta} \rightarrow (j\omega)^{\beta}$$

The fractional derivative models are available for frequency domain analyses only.

Thus, the stress–strain relationship reads

$$\mathrm{dev}(\sigma) = 2\frac{Gp(j\omega)^{\beta}}{G + p(j\omega)^{\beta}}\varepsilon$$

For a standard spring-dashpot branch, the relaxation time is given by the ratio of viscosity and stiffness, $\tau = \eta/G$. Equivalently, the relaxation time τ for a branch with a spring and a spring-pot element in series is derived from

$$\tau^{\beta} = \frac{p}{G}$$

Thus, $p = \tau^{\beta}/G$, and the stress–strain relationship in the spring-spring-pot system reads

$$\mathrm{dev}(\sigma) = 2\frac{G(j\omega\tau)^{\beta}}{1 + (j\omega\tau)^{\beta}}\varepsilon_{\mathrm{d}}$$

The complex-valued shear modulus for this Maxwell element reads

$$G_{\mathrm{M}} = \frac{(j\omega\tau)^{\beta}}{1 + (j\omega\tau)^{\beta}}G$$

Storage and loss moduli are defined as the real and imaginary part of the complex shear modulus

$$G' = \text{real}(G_M) \text{ and } G'' = \text{imag}(G_M)$$

The loss factor is computed from

$$\eta_M = \frac{G''}{G'} = \frac{\sin \beta\pi/2}{(\omega\tau)^\beta + \cos\beta\pi/2}$$

For The Generalized Maxwell Model with fractional derivatives, all dashpots are replaced with spring-pot elements. The rheological representation of this material model is shown in Figure 3-9

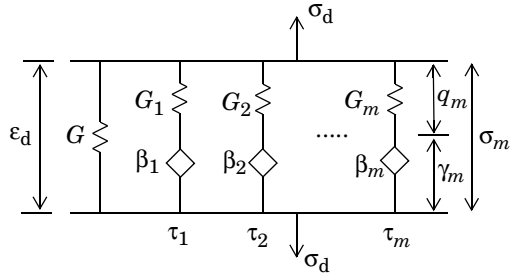


Figure 3-9: The generalized Maxwell model with fractional derivatives.

The complex-valued shear modulus for the generalized Maxwell model with fractional derivatives reads

$$G_{GM} = G + \sum_m \frac{(j\omega\tau_m)^{\beta_m}}{1 + (j\omega\tau_m)^{\beta_m}} G_m$$

where G_m, τ_m, and β_m are the shear modulus, relaxation time, and fractional order of branch m, respectively. Storage and loss moduli are defined as the real and imaginary part of the complex shear modulus.

$$G' = \text{real}(G_{GM}) \text{ and } G'' = \text{imag}(G_{GM})$$

GENERALIZED KELVIN–VOIGT MODEL WITH FRACTIONAL DERIVATIVES

The Kelvin–Voigt viscoelastic model with fractional time derivative consists of a spring connected in parallel with a spring-pot element. The stress strain relation for such arrangement reads

$$\mathrm{dev}(\sigma) = 2pD^{\beta}\varepsilon_d + 2G\varepsilon_d$$

The rheological representation of this arrangement is shown in Figure 3-10

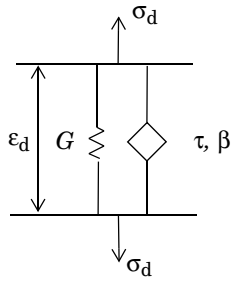


Figure 3-10: Spring spring-pot elements connected in parallel.

In frequency domain, substitute the fractional time derivative operator by

$$D^{\beta} \rightarrow (j\omega)^{\beta}$$

The stress–strain relationship for this Kelvin element then reads

$$\mathrm{dev}(\sigma) = 2(p(j\omega)^{\beta} + G)\varepsilon_d$$

The relaxation time τ for a spring and a spring-pot element in parallel is derived from

$$\tau^{\beta} = \frac{p}{G}$$

so $p = \tau^{\beta}/G$, and the stress–strain relationship in the Kelvin–Voigt element reads

$$\mathrm{dev}(\sigma) = 2G((j\omega\tau)^{\beta} + 1)\varepsilon_d$$

The complex-valued shear modulus in a spring-spring-pot arrangement reads

$$G_{KV} = (1 + (j\omega\tau)^{\beta})G$$

Storage and loss moduli are defined as the real and imaginary part of the complex shear modulus.

$$G' = \mathrm{real}(G_{KV}) = (1 + (\omega\tau)^{\beta}\cos\beta\pi/2)G$$

and

$$G'' = \text{imag}(G_{\text{KV}}) = (\omega\tau)^{\beta}\sin(\beta\pi/2)G$$

The loss factor is computed from

$$\eta_{\text{KV}} = \frac{G''}{G'} = \frac{(\omega\tau)^{\beta}\sin\beta\pi/2}{1+(\omega\tau)^{\beta}\cos\beta\pi/2}$$

The complex-valued compliance of the *Kelvin–Voigt element* with fractional time derivative reads

$$J_{\text{KV}} = \frac{1}{G_{\text{KV}}} = \frac{1}{G}\frac{1}{1+(j\omega\tau)^{\beta}}$$

For The Generalized Kelvin–Voigt Model with fractional derivatives, all dashpots are replaced with spring-pot elements.

The rheological representation of this material model is shown in Figure 3-11.

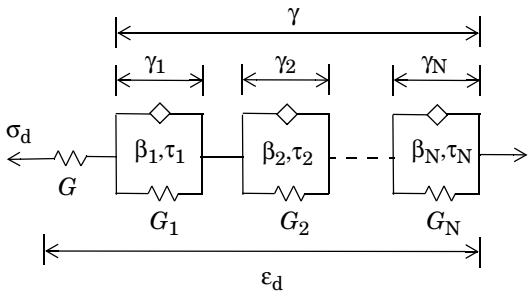


Figure 3-11: The generalized Kelvin–Voigt model with fractional derivatives.

The complex-valued compliance reads

$$J_{\text{GKV}} = \frac{1}{G} + \sum_{m}\frac{1}{1+(j\omega\tau_m)^{\beta_m}}\frac{1}{G_m}$$

where G_m, τ_m, and β_m are the shear modulus, relaxation time, and fractional order of element m, respectively. Storage and loss moduli are defined as the real and imaginary part of the complex shear modulus

$$G_{\text{GKV}} = \frac{1}{J_{\text{GKV}}}$$

$$G' = \text{real}(G_{GKV}) \text{ and } G'' = \text{imag}(G_{GKV})$$

STANDARD LINEAR SOLID MODEL WITH FRACTIONAL DERIVATIVES

In the rheological representation of the Standard Linear Solid Model with fractional derivatives, the dashpot is replaced with a spring-pot element.

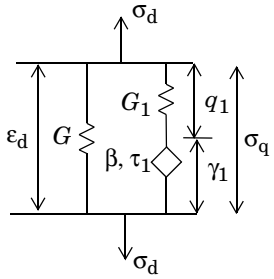


Figure 3-12: Standard linear solid (SLS) model with fractional derivatives.

The deviatoric stress in the spring-spring-pot branch is computed as

$$\sigma_q = 2G_1(\varepsilon_d - \gamma_1) = 2pD^{\beta}\gamma_1$$

here, γ_1 is the strain in the spring-pot element. In the frequency domain this equation translates to

$$\sigma_q = 2p(j\omega)^{\beta}\gamma_1 = \frac{G_1}{G_1 + p(j\omega)^{\beta}} 2p(j\omega)^{\beta}\varepsilon_d$$

Using the relaxation time $\tau = (p/G_1)^{1/\beta}$, this reads

$$\sigma_q = 2\frac{G_1}{1 + (j\omega\tau)^{\beta}}(j\omega\tau)^{\beta}\varepsilon_d = 2G_q\varepsilon_d$$

where $G_q = G_1(j\omega\tau)^{\beta}/(1+(j\omega\tau)^{\beta})$ is the complex shear modulus of the spring-pot branch.

Subsequently, the shear modulus is for the SLS model with fractional derivatives reads

$$G_{SLS} = G + G_q = G + \frac{(j\omega\tau)^{\beta}}{1 + (j\omega\tau)^{\beta}} G_1$$

The storage and loss moduli are defined as the real and imaginary parts of the shear modulus G_{SLS}, respectively

$$G' = \mathrm{real}(G_{\mathrm{SLS}}) \text{ and } G'' = \mathrm{imag}(G_{\mathrm{SLS}})$$

The formulation of this models in time domain requires the use of fractional derivatives. Furthermore, when used in an eigenfrequency analysis, the corresponding fractional order differential equations will always lead to a nonlinear eigenvalue problem.

As an alternative, COMSOL Multiphysics approximates the complex shear modulus in the frequency domain as

$$G_{\mathrm{SLS}}(\omega) \approx G + \sum_{m=1}^{N} \frac{j\omega\tau_m}{1+j\omega\tau_m} G_m$$

where the relaxation times τ_m and the corresponding shear moduli G_m are computed for a given frequency range and accuracy using the partial fraction fitting algorithm. Such approach allows to reuse the implementation of the The Generalized Maxwell Model which is directly suitable for computations is time domain and for eigenfrequency analysis using a linear solver.

BURGERS MODEL WITH FRACTIONAL DERIVATIVES

In the rheological representation of The Burgers Model with fractional derivatives, the dashpots are replaced with a spring-pot elements. The rheological representation of this material model is shown in Figure 3-13:

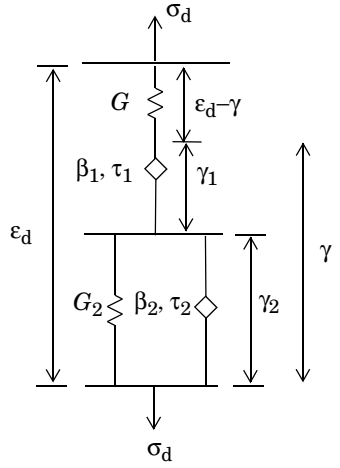


Figure 3-13: The Burgers model with fractional derivatives.

The deviatoric stress in the main branch is computed as

$$\sigma_d = 2G(\varepsilon_d - \gamma) = 2p_1 D^{\beta_1}\gamma_1$$

here, γ_1 is the strain in the spring-pot element. In frequency domain, and using the relaxation times $\tau_1 = (p_1/G)^{1/\beta 1}$, this equation translates to

$$\frac{\sigma_d}{2G} = \varepsilon_d - \gamma = (j\omega\tau_1)^{\beta_1}\gamma_1$$

The stress in the second spring-pot follows

$$2p_2 D^{\beta_2}\gamma_2 = \sigma_d - 2G_2\gamma_2$$

here, γ_2 is the strain in the second spring-pot element. In frequency domain, and using the relaxation time $\tau_2 = (p_2/G_2)^{1/\beta 2}$, this equation reads

$$(j\omega\tau_2)^{\beta_2}\gamma_2 = \frac{\sigma_d}{2G_2} - \gamma_2$$

The total strain in the spring-pots is computed from

$$\gamma = \gamma_1 + \gamma_2$$

which combined with the relations for the strains in the spring-pots

$$\gamma_1 = \frac{1}{2G(j\omega\tau_1)^{\beta_1}}\sigma_d \text{ and } \gamma_2 = \frac{1}{2G_2(1+(j\omega\tau_2)^{\beta_2})}\sigma_d$$

gives the stress and strain relation for Burgers model with fractional derivatives

$$\left(\frac{1}{2G} + \frac{1}{2G(j\omega\tau_1)^{\beta_1}} + \frac{1}{2G_2(1+(j\omega\tau_2)^{\beta_2})}\right)\sigma_d = \varepsilon_d$$

Subsequently, the compliance for the Burgers model with fractional derivatives reads

$$J_B = \frac{1}{G} + \frac{1}{(j\omega\tau_1)^{\beta_1}G} + \frac{1}{(1+(j\omega\tau_2)^{\beta_2})G_2}$$

The storage and loss moduli are defined as the real and imaginary parts of the shear modulus $G_B = 1/J_B$, respectively

$$G' = \text{real}(G_B) \text{ and } G'' = \text{imag}(G_B)$$

USER DEFINED

With the user-defined viscoelastic material model it is possible to directly enter complex-valued expressions for the bulk and shear moduli or compliances, or for the loss factor. The complex-valued bulk and shear moduli are entered in terms of storage and loss moduli or compliances, respectively. The expressions can be functions taken directly from interpolated data, or can be analytical expressions of the frequency variable ω.

When the viscoelastic strain is deviatoric, the stress is computed from the shear storage modulus and shear loss modulus as

$$\sigma_d = 2(G'(\omega) + jG''(\omega))\varepsilon_d$$

or from the storage and loss compliances as

$$\sigma_d = 2\frac{1}{J'(\omega) + jJ''(\omega)}\varepsilon_d$$

When the viscoelastic strain is volumetric, the pressure is computed from the bulk storage modulus and bulk loss modulus

$$p = -(K'(\omega) + jK''(\omega))\varepsilon_{\text{vol}}$$

or from the storage and loss compliances as

$$p = -\frac{1}{Q'(\omega) + jQ''(\omega)}\varepsilon_{\text{vol}}$$

When loss factor damping is selected, the stress–strain relation is augmented by a complex constitutive matrix

$$D^{\text{c}} = (1 + j\eta_{\text{v}}(\omega))D$$

where D is the constitutive matrix computed from the parent material, and D^{c} is the complex constitutive matrix used to compute the viscoelastic stresses.

For time-dependent and eigenfrequency study types, the complex valued elasticity modulus contribution can be approximated in the frequency domain as

$$G_{\text{q}} = G'(\omega) + jG''(\omega) \approx \sum_{m=1}^{N} \frac{j\omega\tau_m}{1 + j\omega\tau_m} G_m$$

where the relaxation times τ_m and the corresponding shear moduli G_m are computed for a given frequency range and accuracy using the partial fraction fitting algorithm. Such approach allows to reuse the implementation of the The Generalized Maxwell Model which is directly suitable for computations is time domain and for eigenfrequency analysis using a linear solver.

TEMPERATURE EFFECTS

For many polymers, the viscoelastic properties have a strong dependence on the temperature. A common assumption is that the material is *thermorheologically simple* (*TRS*). In a material of this class, a change in the temperature can be transformed directly into a change in the time scale. The reduced time is defined as

$$t_r = \int_0^t \frac{dt'}{\alpha_T(T(t'))}$$

where $\alpha_T(T)$ is a temperature-dependent *shift function*.

The implication is that the problem can be solved using the original material data, provided that the time variable is transformed into the reduced time.

Think of the shift function $\alpha_T(T)$ as a multiplier to the viscosity in the dashpot in the Generalized Maxwell model. This multiplier shifts the relaxation time, so Equation 3-34 for a TRS material is modified to

$$\alpha_T(T)\tau_m\dot{\gamma}_m + \gamma_m = \varepsilon_d$$

for all branches.

For the SLS model, the shift applies to the single relaxation time

$$\alpha_T(T)\tau_1\dot{\gamma}_1 + \gamma_1 = \varepsilon_d$$

and for the Kelvin–Voigt model, the shift multiplies the right hand side in Equation 3-39. In general, the shift function α_T will multiply all the relaxation times or viscosities in a given viscoelasticity model.

Williams–Landel–Ferry Shift

One commonly used shift function is defined by the WLF (Williams–Landel–Ferry) equation:

$$\log(\alpha_T) = \frac{-C_1(T - T_{\text{ref}})}{C_2 + (T - T_{\text{ref}})}$$

where a base-10 logarithm is used. This shift is only valid over a certain temperature range, typically around the *glass transition* temperature.

The first step to compute the shift function $\alpha_T(T)$ consists of building a *master curve* based on experimental data. To do this, the curves of the viscoelastic properties (shear modulus, Young's modulus, and so forth) versus time or frequency are measured at a reference temperature T_{ref}. Then, the same properties are measured at different temperatures.

The shift value of each curve, with respect to the master curve obtained at the temperature T_{ref}, defines the shift factor $\alpha_T(T)$. The constants C_1 and C_2 are material dependent and are calculated after plotting $\log(\alpha_T)$ versus $T - T_{\text{ref}}$.

The shift factor at the reference temperature equals $\alpha_T(T_{\text{ref}}) = 1$, so that T_{ref} is the temperature at which the master curve is given. If the temperature T drops below $T_{\text{ref}} - C_2$, the WLF equation is no longer valid.

Since the master curve is measured at an arbitrary reference temperature T_{ref}, the shift function $\alpha_T(T)$ can be derived with respect to any temperature, and it is commonly taken as the shift with respect to the glass transition temperature. The values $C_1 = 17.4$ and $C_2 = 51.6$ K are reasonable approximations for many polymers at this reference temperature.

Arrhenius Shift

Below the *Vicat softening temperature*, the shift factor in polymers is normally assumed to follow an Arrhenius law. In this case, the shift factor is given by the equation

$$\log(\alpha_T) = \frac{Q}{R}\left(\frac{1}{T} - \frac{1}{T_{\text{ref}}}\right)$$

here, a base-e logarithm is used, Q is the activation energy (SI unit: J/mol), and R is the universal gas constant.

Tool–Narayanaswamy–Moynihan Shift

Structural relaxation in glass can be modeled using the so-called Tool–Narayanaswamy–Moynihan shift factor which is given as

$$\log(\alpha_T) = \frac{Q}{R}\left(\frac{\chi}{T} + \frac{1-\chi}{T_{\text{f}}} - \frac{1}{T_{\text{ref}}}\right)$$

here, a base-e logarithm is used, Q is the activation energy (SI unit: J/mol), R is the universal gas constant, T is the current temperature, T_{ref} is a reference temperature, χ is a dimensionless activation energy fraction, and T_{f} is the so-called *fictive temperature*. The fictive temperature is given as the weighted average of partial fictive temperatures.

$$T_f = \sum_i w_i T_{fi} \text{ with } \sum_i w_i = 1$$

Here, w_i are the weights and T_{fi} are the partial fictive temperatures. The partial fictive temperatures are determined from a system of coupled ordinary differential equations (ODEs) which follow Tool's equation

$$\frac{d}{dt} T_{fi} = \frac{T - T_{fi}}{\alpha_T \lambda_{0i}}$$

here, λ_{0i} is a structural relaxation time.

STATIONARY ANALYSIS

For stationary analysis it is possible to select either the long-term stiffness, in which case the stiffness of the viscoelastic branches is neglected, or the instantaneous stiffness, in which case the contribution from all branches is used.

The instantaneous shear modulus G_0 is defined as the sum of the stiffness of all the branches

$$G_0 = G + \sum_{m=1}^{N} G_m$$

FREQUENCY DOMAIN ANALYSIS AND DAMPING

For Frequency-Domain Studies, Equation 3-32 and Equation 3-34 are simplified to

$$\sigma_d = 2(G' + jG'')\varepsilon_d$$

where the *shear storage modulus* G' and the *shear loss modulus* G'' are user defined expressions or are defined for the generalized Maxwell model as

$$G' = G + \sum_{m=1}^{N} G_m \frac{(\omega\tau_m)^2}{1 + (\omega\tau_m)^2} \text{ and } G'' = \sum_{m=1}^{N} G_m \frac{\omega\tau_m}{1 + (\omega\tau_m)^2}$$

for the SLS model as

$$G' = G + G_1 \frac{(\omega\tau_1)^2}{1 + (\omega\tau_1)^2} \text{ and } G'' = G_1 \frac{\omega\tau_1}{1 + (\omega\tau_1)^2}$$

for the Kelvin–Voigt model as

$$G' = G \text{ and } G'' = \eta\omega = G\tau\omega$$

The loss tangent is then computed from the ratio between the loss and storage moduli

$$\tan\delta = \frac{G''}{G'}$$

where δ is the phase angle between deviatoric stress and strain.

The internal work of viscous forces averaged over a time period $2\pi/\omega$ is computed as

$$Q_h = \omega G''\varepsilon_d : \text{conj}(\varepsilon_d)$$

EIGENFREQUENCY ANALYSIS

The direct use of frequency-dependent shear or bulk moduli, as described in Frequency Domain Analysis and Damping, leads to a nonlinear eigenvalue problem. COMSOL Multiphysics solves such nonlinear problems by expanding all nonlinear expressions with quadratic polynomials around a *frequency linearization point* which you can specify in the **Eigenvalue Solver** node. The linearized eigenvalue problem then reads

$$[\mathbf{K}(\omega_L) - \omega^2\mathbf{M}]\mathbf{u} = 0$$

where ω_L is the angular frequency at the linearization point. Thus, the solution of the linearized system of equations depends on the choice of the linearization frequency ω_L.

Selecting the linearization frequency closer to one of the eigenfrequencies will produce a better result for that particular frequency, but not for all eigenfrequencies in the study. Hence, the eigenfrequencies need to be computed one by one using a certain iterative process that updates the linearization frequency ω_L. This iterative procedure is needed when fractional derivatives are used, and also for the User Defined viscoelasticity model.

In order to avoid solving a nonlinear eigenvalue problem for the built-in viscoelasticity models, the same procedure as for time-dependent analysis is used; that is, each viscoelastic branch is represented by additional variables. For instance, for The Generalized Maxwell Model in an eigenfrequency analysis, Equation 3-34 reads

$$j\omega\tau_m\gamma_m + \gamma_m = \varepsilon$$

The viscoelastic strain variables, γ_m, are treated as additional degrees of freedom (DOFs), and contribute to the damping and stiffness matrices of the coupled system of equations.

This procedure results in a damped linear eigenvalue problem that can be solved using the default eigenfrequency solver in a single run, for any specified number of eigenfrequencies.

	For more information see the Eigenfrequency section in the Studies and Solvers chapter in the *COMSOL Multiphysics Reference Manual.*

	An example of a nonlinear eigenvalue problem is shown in *Eigenmodes of a Viscoelastic Structural Damper*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_eigenmodes**

LOCAL TIME INTEGRATION

For The Generalized Maxwell Model and the Standard Linear Solid Model it is possible to use a local time integration algorithm for time-dependent analysis. By using this algorithm, the degrees-of-freedom related to viscoelasticity are treated as internal state variables, which makes the overall solution more efficient and also leaner in terms of memory usage. The implemented algorithm is based on the method suggested in Ref. 28, and it is schematically described in the following.

Consider the time evolution of a viscoelastic branch in the Generalized Maxwell model characterized by an ODE of the type

$$\tau_m \dot{\gamma}_m + \gamma_m = \varepsilon(t) \tag{3-40}$$

where γ_m represents the viscoelastic strain in the m branch. The exact solution to Equation 3-40 is then

$$\gamma_m^{n+1} = \exp\left(-\frac{\Delta t}{\tau_m}\right)\gamma_m^n + \frac{1}{\tau_m}\int_{t^n}^{t^{n+1}} \exp\left(-\frac{t^{n+1}-s}{\tau_m}\right)\varepsilon(s)ds \tag{3-41}$$

By assuming that the strain $\varepsilon(t)$ varies linearly within time increments, the integral in Equation 3-41 can be computed analytically, so the viscoelastic strain at increment $n+1$ reads

$$\gamma_m^{n+1} = \varepsilon^{n+1} - \exp\left(-\frac{\Delta t}{\tau_m}\right)(\varepsilon^n - \gamma_m^n) - \frac{1 - \exp\left(-\frac{\Delta t}{\tau_m}\right)}{\frac{\Delta t}{\tau_m}}(\varepsilon^{n+1} - \varepsilon^n) \qquad (3\text{-}42)$$

All variables in Equation 3-42 are evaluated at the increment n and are stored as internal state variables. Equation 3-42 is solved for each branch in the Generalized Maxwell model, or for the single branch in the Standard Linear Solid model.

See also the description of Viscoelasticity in the Solid Mechanics interface documentation.

Large Strain Viscoelasticity

The implementation for large strain viscoelasticity follows the derivation by Holzapfel (Ref. 1).

The generalized Maxwell model is based on the splitting of the strain energy density into volumetric, isochoric, and the contribution from the viscoelastic branches

$$W_s = W_{\text{iso}} + W_{\text{vol}} + \sum_{m=1}^{N} \psi_m$$

The strain energy in the main hyperelastic branch is normally denoted with the superscript ∞ to denote the long-term equilibrium (as $t \to \infty$).

The second Piola–Kirchhoff stress is computed from

$$S = 2\frac{\partial W_s}{\partial C} = S_{\text{iso}} + S_{\text{vol}} + \sum_{m=1}^{N} Q_m$$

where the auxiliary second Piola–Kirchhoff stress tensors Q_m are defined as

$$Q_m = 2\frac{\partial \psi_m}{\partial C}$$

The time evolution of the auxiliary stress tensor Q_m in each viscoelastic branch is given by the rate

$$\dot{\boldsymbol{Q}}_m + \frac{1}{\tau_m}\boldsymbol{Q}_m = \dot{\boldsymbol{S}}_{\mathrm{iso},m}$$

here, $\boldsymbol{S}_{\mathrm{iso},m}$ is the isochoric second Piola–Kirchhoff stress tensor in the branch m. These tensors are derived from the strain energy density in the main hyperelastic branch and the *energy factors* β_m as

$$\boldsymbol{S}_{\mathrm{iso},\,m} = 2\frac{\partial W_{\mathrm{iso},\,m}}{\partial \boldsymbol{C}} = 2\beta_m \frac{\partial W_{\mathrm{iso}}}{\partial \boldsymbol{C}} = \beta_m \boldsymbol{S}_{\mathrm{iso}}$$

so the time evolution of the auxiliary stress tensor $\boldsymbol{Q}_m$ is given by

$$\dot{\boldsymbol{Q}}_m + \frac{1}{\tau_m}\boldsymbol{Q}_m = \beta_m \dot{\boldsymbol{S}}_{\mathrm{iso}}$$

This equation is not well suited for modeling prestressed bodies. Applying the change of variables

$$\boldsymbol{q}_m = \beta_m \boldsymbol{S}_{\mathrm{iso}} - \boldsymbol{Q}_m$$

the time evolution of the auxiliary stress tensor $\boldsymbol{q}_m$ reads

$$\tau_m \dot{\boldsymbol{q}}_m + \boldsymbol{q}_m = \beta_m \boldsymbol{S}_{\mathrm{iso}}$$

For The Generalized Maxwell Model, it is possible to write the Prony series in terms of weights w_m with respect to the instantaneous stiffness G_0

$$\Gamma(t) = G_0\left(w_\infty + \sum_{m\,=\,1}^{N} w_m \exp\left(-\frac{t}{\tau_m}\right)\right)$$

Here, w_∞ is the ratio between the long term and instantaneous stiffness, $w_\infty = G_\infty/G_0$, and the weight w_m relates the branch stiffness to the instantaneous stiffness through $w_m = G_m/G_0$. The weights are bounded by

$$w_\infty + \sum_{m\,=\,1}^{N} w_m = 1$$

Use this expression to relate the instantaneous stiffness in the small-strain regime to the large strain formulation. The instantaneous response for the second Piola–Kirchhoff stress is computed from

$$S = S_{\text{iso}} + S_{\text{vol}} + \sum_{m=1}^{N} \beta_m S_{\text{iso}}$$

In this expression, the isochoric response represents the long term stiffness, and the sum of the energy factors the contribution to the overall stiffness from the Maxwell branches.

The weights in the Generalized Maxwell model for the small-strain formulation are related to the energy factors in the large strain formulation by

$$\beta_m = w_m \frac{G_0}{G_\infty} = \frac{G_m}{G_\infty}$$

equivalently, the energy factors represent the ratio between the branch stiffness and the long term stiffness in the main branch.

The volumetric contribution from the auxiliary stress tensor q_m is removed before adding these to the viscoelastic stresses per branch $Q_m = \beta_m S_{\text{iso}} - q_m$, so the stresses Q_m are isochoric.

TEMPERATURE EFFECTS

The same options for defining Temperature Effects as described for Linear Viscoelasticity are available for large strain viscoelasticity.

Hyperelastic Materials

A hyperelastic material is defined by its elastic strain energy density W_s, which is a function of the elastic strain state. It is often referred to as the *energy density*. The hyperelastic formulation normally gives a nonlinear relation between stress and strain, as opposed to Hooke's law in linear elasticity.

Most of the time, the *right Cauchy–Green deformation tensor* C is used to describe the current state of strain (although one could use the left Cauchy–Green tensor B, the deformation gradient tensor F, and so forth), so the strain energy density is written as $W_s(C)$.

For isotropic hyperelastic materials, any state of strain can be described in terms of three independent variables — common choices are the *invariants* of the right

Cauchy–Green tensor C, the invariants of the Green–Lagrange strain tensor, or the *principal stretches*.

Once the strain energy density is defined, the second Piola–Kirchhoff stress is computed as

$$S = 2\frac{\partial W_s}{\partial C}$$

In the general case, the expression for the energy W_s is symbolically evaluated down to the components of C using the invariants definitions prior to the calculations of the components of the second Piola–Kirchhoff stress tensor. The differentiation is performed in components on the local coordinate system.

In Equation View, the definitions of the stress components are shown as `solid.Sl11 = 2*d(solid.Ws,solid.Cl11)`, `solid.Sl12 = d(solid.Ws,solid.Cl12)`, and so on. The factor 2 in front of the differentiation operator for the shear stresses is omitted, since the symmetry in the Cauchy–Green tensor will cause two equal contributions.

Modeling Geometric Nonlinearity

For hyperelastic materials, the decomposition between elastic and inelastic deformation is made using a multiplicative decomposition of the deformation gradient

$$F_{el} = FF_{in}^{-1}$$

Here, the inelastic deformation tensor F_{in} depends on the inelastic process, such as thermal expansion, hygroscopic swelling, or plasticity.

In this case, the strain energy density depends on the elastic deformation only, $W_s(C_{el})$, and the second Piola–Kirchhoff stress can be written in terms of its elastic counterpart

$$S = J_{in}F_{in}^{-1}S_{el}F_{in}^{-T}$$

where $J_{in} = \det(F_{in})$ and

$$S_{el} = 2\frac{\partial W_s}{\partial C_{el}}$$

THERMAL EXPANSION AND THERMOELASTIC DAMPING

If thermal expansion is present, a stress-free volume change occurs. This is a pure volumetric change, so the multiplicative decomposition of the deformation gradient tensor in Equation 3-4 implies

$$J_{el} = \frac{\det(F)}{\det(F_{th})} = \frac{J}{J_{th}}$$

Here, the thermal volume ratio, J_{th}, depends on the thermal stretch λ_{th}, which for linear thermal expansion in isotropic materials can be written in terms of the isotropic coefficient of thermal expansion, α_{iso}, and the absolute change in temperature

$$J_{th} = \lambda_{th}^3 \text{ and } \lambda_{th} = 1 + \alpha_{iso}(T - T_{ref})$$

Here, the term $\alpha_{iso}(T - T_{ref})$ is the *thermal strain*. The isotropic thermal gradient is therefore a diagonal tensor defined as

$$F_{th} = \lambda_{th} I$$

The internal variables for the thermal stretch and the thermal volume ratio are named `solid.stchth` and `solid.Jth`.

When the coefficient of thermal expansion α is anisotropic, the thermal strain is computed from

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

and the anisotropic thermal gradient is defined as

$$F_{th} = I + \varepsilon_{th}$$

The tangent coefficient of thermal expansion can be computed then as

$$\alpha_t = F_{th}^{-1}\frac{\partial F_{th}}{\partial T}$$

On the opposite, when the tangent coefficient of thermal expansion is specified as a function of temperature, the thermal strain can be found as

$$\varepsilon_{\text{th}} = \exp(H_{\text{th}}) - I$$

where a matrix exponent is assumed, and the Hencky thermal strain tensor is defined as

$$H_{\text{th}} = \int_{T_{\text{ref}}}^{T} \alpha_{\text{t}}(\tau) d\tau$$

When the thermoelastic damping effect needs to be taken into account, the heat source term can be computed as

$$Q_{\text{ted}} = -T \frac{\partial M}{\partial t} : \alpha_{\text{t}}$$

where the Mandel stress tensor is introduced as $M = CS$.

HYGROSCOPIC SWELLING

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain depends linearly on the moisture content

$$\varepsilon_{\text{hs}} = \beta_{\text{h}}(c_{\text{mo}} - c_{\text{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\text{mo,ref}}$ is the strain-free reference concentration. The coefficient of hygroscopic swelling can represent isotropic or anisotropic swelling. The anisotropic hygroscopic gradient is defined as

$$F_{\text{hs}} = I + \varepsilon_{\text{hs}}$$

SHRINKAGE AND SWELLING

Shrinkage and swelling strains are caused by transport of species in solids. The volumetric strain is written as

$$\varepsilon_{\text{sas,vol}} = \Omega_{\text{Vc}}(c - c_{\text{ref}})$$

where Ω_{V} is the volumetric coefficient of swelling, c is the species concentration, and c_{ref} is the strain-free reference concentration. c_i is the species that diffuses in the solid. It can have different units based on the user input (SI unit: kg/m^3, mol/m^3, or $1/\text{m}^3$)

For a large strain formulation, the volume change reads

$$J_{\text{sas}} = 1 + \Omega_{\text{Vc}}(c - c_{\text{ref}})$$

When the linear coefficient of swelling is used, the strain tensor resulting from swelling and shrinkage reads

$$\varepsilon_{sas} = \Omega_c(c - c_{ref})$$

and for a large strain formulation, the deformation gradient reads

$$F_{sas} = I + \Omega_c(c - c_{ref})$$

where the linear coefficient of swelling, Ω_c, is an isotropic or diagonal tensor. Note that for an isotropic tensor, its trace corresponds to the volumetric coefficient of swelling

$$\text{trace}(\Omega_c) = 3\Omega_c = \Omega_{Vc}$$

The shrinkage and swelling effect is bidirectional, since there is a flux contribution to The Transport Equation given by gradients in the stress field.

When M is the mobility and μ_{el} is the potential due to the strain energy density, the stress-driven flux in transport equation reads

$$\Gamma_{ext} = -M\nabla\mu_{el}$$

where the potential is derived from the strain energy density W_s

$$\mu_{el} = \frac{\partial W_s}{\partial c}$$

For an isotropic linear elastic material and using a linear Mobility relation, this flux reads

$$\Gamma_{ext} = -M\Omega_{Vc}\nabla p = -\frac{Dc}{RT}\Omega_{Vc}\nabla p$$

The flux vector Γ in the transport equation contains both the flux due to Fickian diffusion and the flux due to the pressure gradient

$$\Gamma_i = \Gamma_{dif} + \Gamma_{ext}$$

HYPERELASTICITY WITH PLASTICITY

It is possible to combine the hyperelastic material models with plasticity, viscoplasticity, and creep. Since these models are primarily used for large strain applications, only the large strain formulation is available. The decomposition between elastic and plastic deformation is made using a multiplicative decomposition of the deformation gradient tensor,

$$F_{\text{el}} = FF_{\text{pl}}^{-1}$$

Here, the plastic deformation tensor F_{pl} depends on the plastic flow rule, yield function, and plastic potential.

- Multiplicative Decomposition
- Multiplicative Plastic Flow
- Creep and Viscoplasticity for Large Strains

ISOCHORIC ELASTIC DEFORMATION

For some classes of hyperelastic materials it is convenient to split the strain energy density into *volumetric* (also called *dilatational*) and *isochoric* (also called *distortional* or *volume preserving*) contributions. The elastic deformation tensor is then multiplicatively decomposed into the volumetric and isochoric components

$$F_{\text{el}} = F_{\text{el,vol}}\overline{F}_{\text{el}}$$

with $F_{\text{el,vol}}$ as the volumetric elastic deformation (a diagonal tensor) and $\overline{F}_{\text{el}}$ the isochoric elastic deformation gradient. Isochoric deformation means that the volume ratio is kept constant during deformation, so the isochoric elastic deformation is computed by scaling it by the elastic volume ratio. The elastic volume ratio is defined by

$$J_{\text{el}} = \det(F_{\text{el}}) = \det(F_{\text{el,vol}})$$

and the volumetric deformation as

$$F_{\text{el,vol}} = J_{\text{el}}^{1/3}I$$

By using J_{el} it is possible to define the *isochoric-elastic deformation* gradient

$$\overline{F}_{\text{el}} = J_{\text{el}}^{-1/3}F_{\text{el}}$$

the *isochoric-elastic right Cauchy–Green* tensor

$$\overline{C}_{\text{el}} = \overline{F}_{\text{el}}^{T}\overline{F}_{\text{el}} = J_{\text{el}}^{-2/3}C_{\text{el}}$$

and the *isochoric-elastic Green–Lagrange strain* tensor

$$\overline{\varepsilon_{el}} = \frac{1}{2}(\overline{C_{el}} - \mathbf{I})$$

This scaling changes the eigenvalues of the tensor, but not its principal directions, so the original and isochoric tensors remain coaxial to each other.

Some authors call $\overline{F_{el}}$ and $\overline{C_{el}}$ the *modified* tensors. Note that

$$\det(\overline{F_{el}}) = \det(\overline{C_{el}}) = 1$$

The internal variables for the isochoric-elastic Cauchy–Green deformation tensor in local coordinate system are named `solid.CIel11`, `solid.CIel12`, and so on.

The other two invariants normally used together with J_{el} are the first and second invariant of the isochoric-elastic right Cauchy–Green deformation tensor

$$I_1(\overline{C_{el}}) \text{ and } I_2(\overline{C_{el}})$$

In these equations:

$$I_1(\overline{C_{el}}) = \text{trace}(\overline{C_{el}}) = J_{el}^{-2/3} I_1(C_{el})$$

$$I_2(\overline{C_{el}}) = \frac{1}{2}(I_1^2(\overline{C_{el}}) - \text{trace}(\overline{C_{el}}^2)) = J_{el}^{-4/3} I_2(C_{el})$$

Since $I_3(\overline{C_{el}}) = \det(\overline{C_{el}}) = 1$, the third invariant is never explicitly used.

The internal variables for the invariants J_{el}, $I_1(\overline{C_{el}})$, and $I_2(\overline{C_{el}})$ are named `solid.Jel`, `solid.I1CIel`, and `solid.I2CIel`.

The invariants of the isochoric (modified) elastic Green–Lagrange strain tensor are related to the invariants of the isochoric-elastic right Cauchy–Green deformation tensor

$$I_1(\overline{\varepsilon_{el}}) = \text{trace}(\overline{\varepsilon_{el}}) = \frac{1}{2}(I_1(\overline{C_{el}}) - 3)$$

$$I_2(\overline{\varepsilon_{el}}) = \frac{1}{2}(I_1^2(\overline{\varepsilon_{el}}) - \text{trace}(\overline{\varepsilon_{el}}^2)) = \frac{1}{4}(I_2(\overline{C_{el}}) - 2I_1(\overline{C_{el}}) + 3)$$

$$I_3(\overline{\varepsilon_{el}}) = \det(\overline{\varepsilon_{el}}) = \frac{1}{8}(I_1(\overline{C_{el}}) - I_2(\overline{C_{el}}))$$

The internal variables for the invariants of the isochoric elastic Green–Lagrange strain tensor are named `solid.I1eIel`, `solid.I2eIel`, and `solid.I3eIel`.

COUPLED AND UNCOUPLED RESPONSES

The isochoric and volumetric responses of most materials are *coupled* (Ref. 12). In this case, the strain energy density is written as a function of the elastic deformation

$$W_s = W_s(C_{el})$$

However, in many situations it is possible to assume that the pressure and deviatoric stress are *uncoupled*, and therefore for numerous hyperelastic materials it is possible to define the strain energy density as the sum of isochoric and volumetric counterparts

$$W_s = W_{iso}(\overline{C_{el}}) + W_{vol}(J_{el})$$

Here, $\overline{C_{el}}$ is the isochoric elastic right Cauchy–Green deformation tensor, and J_{el} is the elastic volume ratio.

VOLUMETRIC RESPONSE

The volumetric strain energy density, W_{vol}, is defined as an expression of the elastic volumetric deformation J_{el} and the bulk modulus κ.

The **Quadratic** volumetric strain energy density is defined as

$$W_{vol}(J_{el}) = \frac{\kappa}{2}(J_{el} - 1)^2 \tag{3-43}$$

The **Polynomial** volumetric strain energy density is defined as a higher-order expansion of the volumetric option:

$$W_{vol}(J_{el}) = \frac{\kappa}{2}(J_{el} - 1)^2 + \frac{\kappa_2}{4}(J_{el} - 1)^4 + \frac{\kappa_3}{6}(J_{el} - 1)^6 \tag{3-44}$$

The **Logarithmic** volumetric strain energy density (Ref. 12) is defined as

$$W_{vol}(J_{el}) = \frac{\kappa}{2}(\ln J_{el})^2 \tag{3-45}$$

The **Hartmann–Neff** volumetric strain energy density (Ref. 13) is defined as

$$W_{\text{vol}}(J_{\text{el}}) = \frac{\kappa}{50}(J_{\text{el}}^5 + J_{\text{el}}^{-5} - 2) \tag{3-46}$$

The **Miehe** volumetric strain energy density (Ref. 15) is defined as

$$W_{\text{vol}}(J_{\text{el}}) = \kappa(J_{\text{el}} - \ln J_{\text{el}} - 1) \tag{3-47}$$

and the **Simo–Taylor** volumetric strain energy density (Ref. 16) is defined as

$$W_{\text{vol}}(J_{\text{el}}) = \frac{\kappa}{4}(J_{\text{el}}^2 - 2\ln J_{\text{el}} - 1) \tag{3-48}$$

It is also possible to enter a **User defined** expression for the volumetric strain energy density. The expression should be a convex function of the elastic volume ratio `solid.Jel`, where `solid` is the name of the physics interface node.

In the Membrane interface, a user-defined volumetric strain energy density must be a function of the *modified* elastic volume ratio `mbrn.Jelw`.

For a given volumetric strain energy density, the volumetric stress (pressure) is calculated as

$$p_{\text{m}} = -\frac{\partial W_{\text{vol}}}{\partial J_{\text{el}}} \tag{3-49}$$

When the quadratic expression in Equation 3-43 is used, the pressure becomes linearly related to the volume change:

$$p_{\text{m}} = -\kappa(J_{\text{el}} - 1)$$

but for other volumetric representations the pressure becomes a nonlinear function of the volume change. For instance, if the expression in Equation 3-46 is used instead of the quadratic function, the pressure reads

$$p_{\text{m}} = -\frac{1}{10}\kappa({J_{\text{el}}}^4 - {J_{\text{el}}}^{-6})$$

THE LOCKING PROBLEM

A numerical scheme is said to exhibit *locking* if the accuracy of the approximation deteriorates as a parameter tends to a limiting value (Ref. 17). Finite elements in solid mechanics are said to "lock" when exhibiting an unphysical response to deformation (Ref. 18). Locking can occur for many different reasons. For linear elastic materials, this typically happens as Poisson's ratio tends to 0.5, or the bulk modulus is much larger than the shear modulus. Numerical errors arise because the shape functions are unable to properly describe the volume preserving deformation.

To avoid the locking problem in computations, the mixed formulation replaces p_{m} in Equation 3-49 with a corresponding interpolated pressure help variable p_{w}, which adds an extra degree of freedom to the ones defined by the displacement vector $\mathbf{u}$.

The general procedure is the same as described in Mixed Formulation for Linear Elastic materials.

INCOMPRESSIBLE HYPERELASTIC MATERIALS

For incompressible hyperelastic materials, the volumetric strain energy density W_{vol} is not defined at all, and the strain energy density W_{s} consists only of the isochoric contribution and the incompressibility constraint

$$W_{\mathrm{s}}(\mathbf{u}, p_{\mathrm{w}}) = W_{\mathrm{iso}} - (J_{\mathrm{el}} - 1)p_{\mathrm{w}}$$

Instead of deriving the pressure from the volumetric strain energy, a weak constraint is added to account for the incompressibility condition

$$J_{\mathrm{el}} = 1$$

The auxiliary pressure variable, p_{w}, acts as Lagrange multiplier to enforce the constraint $J_{\mathrm{el}} = 1$. This variable, positive in compression, is then used as the pressure when computing stresses.

The contribution to the virtual work is

$$\delta W = -\int_V S{:}\delta\varepsilon \, dv + \int_V (J_{\mathrm{el}} - 1)\delta p_{\mathrm{w}} \, dv$$

The second Piola–Kirchhoff stress is then given by

$$S = 2\frac{\partial W_{\mathrm{iso}}}{\partial C} - p_{\mathrm{w}} J C^{-1} \tag{3-50}$$

and the Cauchy stress tensor by

$$\sigma = J^{-1}FSF^T = 2J^{-1}F\frac{\partial W_{\text{iso}}}{\partial C}F^T - p_{\text{w}}I \tag{3-51}$$

NEARLY INCOMPRESSIBLE HYPERELASTIC MATERIALS

For nearly incompressible hyperelastic materials, the strain energy density W_{s} is decoupled into isochoric and volumetric counterparts:

$$W_{\text{s}} = W_{\text{iso}} + W_{\text{vol}}$$

here, W_{iso} is the isochoric strain energy density and W_{vol} is the volumetric strain energy density.

The isochoric strain energy density depends on the material model, and it can be an expression involving the following:

- Components of the isochoric elastic right Cauchy–Green tensor $\overline{C_{\text{el}}}$ in the local coordinate system.
- Principal invariants of the isochoric elastic right Cauchy–Green tensor $\overline{C_{\text{el}}}$.
- Principal invariants of the isochoric elastic Green–Lagrange strain $\overline{\varepsilon_{\text{el}}}$.
- For User Defined hyperelastic materials, it is also possible to define the strain energy density in terms of components of the isochoric elastic deformation gradient $\overline{F_{\text{el}}}$.

The volumetric strain energy density can be any of the expression described in Volumetric Response. The default value for the bulk modulus is set to 100 times the equivalent shear modulus at infinitesimal deformation, which gives an initial Poisson's ratio of approximately $\nu = 0.495$.

An auxiliary variable, p_{w}, is added to map the pressure p_{m} derived from the volumetric strain energy density as described in Equation 3-49.

Then the variational problem is computed by the so-called *perturbed Lagrangian method* (Ref. 19), so the contribution to the virtual work reads

$$\delta W = -\int_V S{:}\delta\varepsilon \, dv + \int_V \frac{1}{\kappa}(p_{\text{w}} - p_{\text{m}})\delta p_{\text{w}} \, dv$$

The second Piola–Kirchhoff stress is then defined as

$$S = 2\frac{\partial W_{\text{iso}}}{\partial C} - p_{\text{w}}JC^{-1} \tag{3-52}$$

where the extra pressure variable satisfies the weak constraint

$$p_w = p_m \text{ where } p_m = -\frac{\partial W_{vol}}{\partial J_{el}}$$

the Cauchy stress tensor then reads

$$\sigma = J^{-1}FSF^T = 2J^{-1}F\frac{\partial W_{iso}}{\partial C}F^T - p_w I$$

!	Using the **Quadratic** volumetric strain energy density results in the only mixed formulation that returns a symmetric coupled stiffness matrix.

PREDEFINED HYPERELASTIC MATERIAL MODELS

Different hyperelastic material models are constructed by specifying different elastic strain energy expressions. There are several predefined material models, as well as the option to enter user defined expressions for the isochoric and volumetric strain energy densities.

NEO-HOOKEAN

The strain energy density for the **Compressible, coupled** version of the Neo-Hookean material is written in terms of the elastic volume ratio J_{el} and the first invariant of the elastic right Cauchy–Green deformation tensor $I_1(C_{el})$ (Ref. 11, 14)

$$W_s = \frac{1}{2}\mu(I_1 - 3) - \mu\ln(J_{el}) + \frac{1}{2}\lambda\ln(J_{el})^2$$

Here, λ and μ are the Lamé parameters (SI unit: Pa).

In Ref. 127 the coupled strain energy density is defined from Lamé parameters

$$W_s = \frac{1}{2}\mu(I_1 - 3) + \frac{\mu}{\beta}(J_{el}^{-\beta} - 1) \text{ with } \beta = \frac{\lambda}{\mu} = \frac{2\nu}{1-2\nu}$$

The **Compressible, uncoupled** and **Nearly incompressible** versions use the isochoric invariant $I_1(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_s = W_{iso} + W_{vol} = \frac{1}{2}\mu(\overline{I}_1 - 3) + W_{vol}$$

where the volumetric strain energy W_{vol} can use any of the expressions described in Volumetric Response. See Nearly Incompressible Hyperelastic Materials for details.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

It is possible to define the isotropic hyperelastic material from other pair of elastic moduli, see the conversion from one set of elastic moduli to Lamé parameters in Table 3-1.

See also the description of the Neo-Hookean material model in the Solid Mechanics interface documentation.

SAINT-VENANT–KIRCHHOFF

One of the simplest hyperelastic material models is the Saint-Venant–Kirchhoff material, which is an extension of a linear elastic material into the hyperelastic regime.

For the **Compressible, coupled** response, the elastic strain energy density is written with two parameters and two invariants of the elastic Green–Lagrange strain tensor, $I_1(\varepsilon_{el})$ and $I_2(\varepsilon_{el})$

$$W_s = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2$$

Here, λ and μ are Lamé parameters (SI unit: Pa). The bulk modulus κ is calculated from $\kappa = \lambda + 2\mu/3$.

The **Compressible, uncoupled** and the **Nearly incompressible** versions use the isochoric invariants $I_1(\overline{\varepsilon_{el}})$ and $I_2(\overline{\varepsilon_{el}})$ to define the isochoric strain energy density

$$W_{iso} = \frac{1}{2}(\lambda + 2\mu)\overline{I_1}^2 - 2\mu\overline{I_2}$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

It is possible to define the isotropic hyperelastic material from other pair of elastic moduli, see the conversion from one set of elastic moduli to Lamé parameters in Table 3-1.

See also the description of the Saint-Venant–Kirchhoff material model in the Solid Mechanics interface documentation.

MOONEY–RIVLIN, TWO PARAMETERS

For the **Compressible, uncoupled** and **Nearly incompressible** versions, the isochoric strain energy density is written in terms of the two isochoric invariants of the elastic right Cauchy–Green deformation tensors $I_1(\overline{C}_{\mathrm{el}})$ and $I_2(\overline{C}_{\mathrm{el}})$

$$W_{\mathrm{iso}} = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3)$$

The material parameters C_{10} and C_{01} (SI unit: Pa) are related to the Lamé parameter (shear modulus) $\mu = 2(C_{10} + C_{01})$.

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{\mathrm{el}} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Mooney–Rivlin, Two Parameters material model in the Solid Mechanics interface documentation.

MOONEY–RIVLIN, FIVE PARAMETERS

Rivlin and Saunders (Ref. 2) proposed a phenomenological model for small deformations in rubber-based materials on a polynomial expansion of the first two invariants of the elastic right Cauchy–Green deformation, so the strain energy density is written as an infinite series

$$W_{\mathrm{s}} = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} C_{mn}(I_1 - 3)^m (I_2 - 3)^n$$

with $C_{00} = 0$. This material model is sometimes also called *polynomial hyperelastic material*.

In the first-order approximation, the material model recovers the Mooney–Rivlin strain energy density

$$W_{\mathrm{s}} = C_{10}(I_1 - 3) + C_{01}(I_2 - 3)$$

while the second-order approximation incorporates second-order terms

$$W_s = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{20}(I_1 - 3)^2 + C_{02}(I_2 - 3)^2 + C_{11}(I_1 - 3)(I_2 - 3)$$

The **Compressible, uncoupled** and the **Nearly incompressible** versions use the isochoric invariants of the elastic right Cauchy–Green deformation tensors $I_1(\overline{C}_{el})$ and $I_2(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_{iso} = \sum_{m=0}^{2} \sum_{n=0}^{2} C_{mn}(\overline{I}_1 - 3)^m (\overline{I}_2 - 3)^n$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The incompressible option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Mooney–Rivlin, Five Parameters material model in the Solid Mechanics interface documentation.

MOONEY–RIVLIN, NINE PARAMETERS

The Mooney–Rivlin, nine parameters material model is an extension of the polynomial expression to third-order terms and the strain energy density is written as

$$W_{iso} = \sum_{m=0}^{3} \sum_{n=0}^{3} C_{mn}(\overline{I}_1 - 3)^m (\overline{I}_2 - 3)^n$$

where $I_1(\overline{C}_{el})$ and $I_2(\overline{C}_{el})$ are the isochoric invariants of the elastic right Cauchy–Green deformation tensors.

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Volumetric Response and Incompressible Hyperelastic Materials.

See also the description of the Mooney–Rivlin, Nine Parameters material model in the Solid Mechanics interface documentation.

YEOH

Yeoh proposed (Ref. 1) a phenomenological model in order to fit experimental data of filled rubbers, where Mooney–Rivlin and Neo-Hookean models were to simple to describe the stiffening effect in the large strain regime. The strain energy was fitted to experimental data by means of three parameters (SI unit: Pa), and the first invariant of the elastic right Cauchy–Green deformation tensors $I_1(C_{el})$

$$W_s = c_1(I_1 - 3) + c_2(I_1 - 3)^2 + c_3(I_1 - 3)^3$$

The **Compressible, uncoupled** and the **Nearly incompressible** versions use the isochoric invariant of the elastic right Cauchy–Green deformation tensor $I_1(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_{iso} = c_1(\overline{I}_1 - 3) + c_2(\overline{I}_1 - 3)^2 + c_3(\overline{I}_1 - 3)^3$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Yeoh material model in the Solid Mechanics interface documentation.

OGDEN

The Neo-Hookean material model usually fits well to experimental data at moderate strains but fails to model hyperelastic deformations at high strains. In order to model rubber-like materials at high strains, Ogden adapted (Ref. 1) the energy of a Neo-Hookean material to

$$W_s = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p}(\lambda_{el1}^{\alpha_p} + \lambda_{el2}^{\alpha_p} + \lambda_{el3}^{\alpha_p} - 3)$$

Here μ_p (SI unit: Pa) and α_p (dimensionless) are material parameters, and λ_{el1}, λ_{el2}, and λ_{el3} are the principal elastic stretches such as $J_{el} = \lambda_{el1}\lambda_{el2}\lambda_{el3}$.

The Ogden model is empirical, in the sense that it does not relate the material parameters α_p and μ_p to physical phenomena. The parameters α_p and μ_p are obtained by curve-fitting measured data, which can be difficult for $N > 2$. The most common implementation of Ogden material is with $N = 2$, so four parameters are needed.

The **Compressible, uncoupled** and the **Nearly incompressible** versions use the isochoric elastic stretches

$$\bar{\lambda}_{eli} = \lambda_{eli} / J_{el}^{1/3}$$

so the isochoric strain energy density is defined as

$$W_{iso} = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p}(\bar{\lambda}_{el1}^{\alpha_p} + \bar{\lambda}_{el2}^{\alpha_p} + \bar{\lambda}_{el3}^{\alpha_p} - 3)$$

The isochoric elastic stretches define a volume preserving deformation, since

$$\bar{\lambda}_{el1}\bar{\lambda}_{el2}\bar{\lambda}_{el3} = \lambda_{el1}\lambda_{el2}\lambda_{el3} / J_{el} = 1$$

The shear modulus at infinitesimal deformation is then defined from

$$\mu = \frac{1}{2}\sum_{p=1}^{N} \alpha_p \mu_p$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

!	Sometimes a slightly different definition of the strain energy function for the Ogden material is used. If you have given material data for an Ogden material, be careful that the definitions used are the same. If not, you will need to rescale the μ_p coefficients.

	See also the description of the Ogden material model in the Solid Mechanics interface documentation.

STORAKERS

The Storakers material (Ref. 18 and Ref. 25) is used to model highly compressible foams. The strain energy density is written in a similar fashion as for the Ogden material:

$$W_s = \sum_{k=1}^{N} \frac{2\mu_k}{\alpha_k^2}\left(\lambda_{el1}^{\alpha_k} + \lambda_{el2}^{\alpha_k} + \lambda_{el3}^{\alpha_k} - 3 + \frac{1}{\beta_k}(J_{el}^{-\alpha_k\beta_k} - 1)\right)$$

The initial shear and bulk moduli are computed from the parameters μ_k (SI unit: Pa) and β_k (dimensionless) as

$$\mu = \sum_{k=1}^{N} \mu_k \quad \text{and} \quad \kappa = \sum_{k=1}^{N} 2\mu_k\left(\beta_k + \frac{1}{3}\right)$$

For constant parameters $\beta_k = \beta$, the initial bulk modulus becomes $\kappa = 2\mu(\beta + 1/3)$, so a stable material requires $\mu > 0$ and $\beta > -1/3$. In this case, the Poisson's ratio is given by $\nu = \beta/(2\beta + 2/3)$, which means that for a Poisson's ratio larger than -1, $\beta > -2/9$ is needed.

	See also the description of the Storakers material model in the Solid Mechanics interface documentation.

VARGA

The Varga material model (Ref. 1) describes the strain energy in terms of the elastic stretches as

$$W_s = c_1(\lambda_{el1} + \lambda_{el2} + \lambda_{el3} - 3) + c_2(\lambda_{el1}\lambda_{el2} + \lambda_{el2}\lambda_{el3} + \lambda_{el1}\lambda_{el3} - 3)$$

The **Compressible, uncoupled** and **Nearly incompressible** versions use the isochoric elastic stretches defined as

$$\bar{\lambda}_{eli} = \lambda_{eli}/J_{el}^{1/3}$$

to define the isochoric strain energy density:

$$W_{iso} = c_1(\bar{\lambda}_{el1} + \bar{\lambda}_{el2} + \bar{\lambda}_{el3} - 3) + c_2(\bar{\lambda}_{el1}\bar{\lambda}_{el2} + \bar{\lambda}_{el2}\bar{\lambda}_{el3} + \bar{\lambda}_{el1}\bar{\lambda}_{el3} - 3)$$

The simplest Varga model is obtained by setting $c_1 = 2\mu$ and $c_2 = 0$:

$$W_{iso} = 2\mu(\bar{\lambda}_{el1} + \bar{\lambda}_{el2} + \bar{\lambda}_{el3} - 3)$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Varga material model in the Solid Mechanics interface documentation.

ARRUDA–BOYCE

The other hyperelastic materials described are phenomenological models in the sense that they do not relate the different material parameters (normally obtained by curve-fitting experimental data) to physical phenomena.

Arruda and Boyce (Ref. 3) derived a material model based on Langevin statistics of polymer chains. The strain energy density is defined by

$$W_s = nk_BTN\left(\frac{\lambda_{chain}}{\lambda_{lock}}\beta + \ln\left(\frac{\beta}{\sinh(\beta)}\right)\right)$$

Here, n is the chain density, k_B is the Boltzmann constant, T is the absolute temperature, and β is computed from the inverse Langevin function $\beta = L^{-1}(\lambda_{chain}/\lambda_{lock})$.

The number of segments in the polymeric chain, N, is used to derive the locking stretch of the chain, $\lambda_{lock} = \sqrt{N}$, and the equivalent chain stretch λ_{chain} is computed from the first invariant of the elastic right Cauchy–Green deformation tensor $I_1(C_{el})$, $\lambda_{chain} = \sqrt{I_1/3}$.

The macroscopic stiffness of the network is defined as $\mu_0 = nk_BT$. This material model is sometimes also called the *eight-chain model*.

In Ref. 3, a series expansion of the inverse Langevin function is used to simplify the expression for the strain energy density. In this case, the strain energy density reads

$$W_s = \mu_0 \sum_{p=1}^{\infty} \frac{c_p}{N^{p-1}} (I_1^p - 3^p)$$

Here, μ_0 is the macroscopic stiffness of the network, $I_1(C_{el})$ is the first invariant of the elastic right Cauchy–Green deformation tensor, and the coefficients c_p are obtained by a series expansion of the inverse Langevin function.

Arruda and Boyce truncated the series and used only the first five terms. These coefficients c_p (dimensionless) are listed in Table 3-2:

TABLE 3-2: FIRST FIVE COEFFICIENTS OF ARRUDA–BOYCE MATERIAL MODEL.

c_1	c_2	c_3	c_4	c_5
1/2	1/20	11/1050	19/7000	519/673750

Other authors (Ref. 1) use only the first three terms of the series.

The **Compressible, uncoupled** and **Nearly incompressible** versions use the isochoric invariant $\overline{I_1} = I_1(\overline{C_{el}})$ to define the isochoric strain energy density, so when using the inverse Langevin function the chain stretch is defined as $\lambda_{chain} = \sqrt{\overline{I_1}/3}$, and in the five terms approximation the isochoric strain energy is defined as

$$W_{iso} = \mu_0 \sum_{p=1}^{5} \frac{c_p}{N^{p-1}} (\overline{I_1}^p - 3^p)$$

Then, the elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Arruda–Boyce material model in the Solid Mechanics interface documentation.

GENT

Many hyperelastic material models are difficult to fit to experimental data. The Gent material (Ref. 20 and Ref. 21) is a simple phenomenological constitutive model based on only two parameters, μ and j_m, which defines the strain energy density as

$$W_s = -\frac{\mu}{2} j_m \log\left(1 - \frac{\overline{I}_1 - 3}{j_m}\right)$$

Here, μ (SI unit: Pa) is the shear modulus at small strains and j_m (dimensionless) is a limiting value for $I_1 - 3$, which accounts for the limiting polymeric chain extensibility of the material.

Since the strain energy density does not depend on the second invariant I_2, the Gent model is often classified as a generalized Neo-Hookean material. The strain energy density approaches that of an incompressible Neo-Hookean material as $j_m \to \infty$. The Gent material is the simplest model in the limiting chain extensibility family.

The **Compressible, uncoupled** and **Nearly incompressible** formulations use the isochoric invariants $I_1(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_{iso} = -\frac{\mu}{2} j_m \log\left(1 - \frac{\overline{I}_1 - 3}{j_m}\right)$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

See also the description of the Gent material model in the Solid Mechanics interface documentation.

VAN DER WAALS

Kilian and co-workers formulated a van der Waals equation of state for real networks of polymer chains from the analogy of an ideal gas (Ref. 22–24). The constitutive model is based on four parameters which define the averaged invariant

$$\bar{I} = \beta \bar{I}_1 + (1-\beta)\bar{I}_2$$

the variable Θ

$$\Theta = \sqrt{\frac{\bar{I}-3}{\lambda_m^2-3}}$$

and the function

$$f(\Theta) = -\log(1-\Theta) - \Theta$$

in order to define the isochoric strain energy density

$$W_{iso} = \mu\left((\lambda_m^2-3)f(\Theta) - \frac{2\alpha}{3}\left(\frac{\bar{I}-3}{2}\right)^{\frac{3}{2}}\right)$$

Here, μ (SI unit: Pa) is the small-strain shear modulus. The maximum chain stretch λ_m (dimensionless) represents a limiting value for the averaged invariant $\bar{I}$, which accounts for the maximum chain extensibility in networks with finite chain lengths. The dimensionless parameter β phenomenologically averages the two isochoric invariants $\bar{I}_1$ and $\bar{I}_2$, and the parameter α (dimensionless) represents the global interaction between polymer chains.

The **Compressible, uncoupled** and **Nearly incompressible** formulations use the elastic volume ratio J_{el} and the bulk modulus κ to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{el} = 1$; see Incompressible Hyperelastic Materials

See also the description of the van der Waals material model in the Solid Mechanics interface documentation.

BLATZ–KO

The Blatz–Ko material model was developed for foamed elastomers and polyurethane rubbers, and it is used to model compressible isotropic hyperelastic materials (Ref. 1, 26).

The elastic strain energy density is written with three parameters and the three invariants of the elastic right Cauchy–Green deformation tensor, $I_1(C_{el})$, $I_2(C_{el})$, and $I_3(C_{el}) = J_{el}^2$

$$W_s = \phi\frac{\mu}{2}\left((I_1 - 3) + \frac{1}{\beta}(I_3^{-\beta} - 1)\right) + (1-\phi)\frac{\mu}{2}\left(\left(\frac{I_2}{I_3} - 3\right) + \frac{1}{\beta}(I_3^{\beta} - 1)\right)$$

Here, ϕ is an interpolation parameter bounded to $0 < \phi < 1$, the parameter μ (SI unit: Pa) is the small-strain shear modulus, and β can be related to the Poisson's ratio, $\beta = \nu/(1 - 2\nu)$, or the Lamé parameters, $\beta = \lambda/(2\mu)$.

When the parameter $\beta \to \infty$, or equivalently, Poisson's ratio tends to 0.5, the strain energy simplifies to a similar form of the Mooney–Rivlin, Two Parameters model

$$W_s = \phi\frac{\mu}{2}(I_1 - 3) + (1-\phi)\frac{\mu}{2}\left(\frac{I_2}{I_3} - 3\right)$$

In the special case of $\phi = 1$, the strain energy reduces to a compressible Neo-Hookean model (use $2\beta = \lambda/\mu = 2\nu/(1 - 2\nu)$),

$$W_s = \frac{\mu}{2}(I_1 - 3) + \frac{\mu}{2\beta}(J^{-2\beta} - 1)$$

This expression is also equivalent to the compressible Storakers model consisting of one term with the exponent $\alpha_1 = 2$.

See also the description of the Blatz–Ko material model in the Solid Mechanics interface documentation.

GAO

Gao proposed (Ref. 27) a simple compressible hyperelastic material where the strain energy density is defined by two parameters, a (SI unit: Pa) and n, and two invariants of the elastic right Cauchy–Green deformation tensors C_{el}:

$$W_s = a(I_1^n + I_{-1}^n)$$

Here, the invariant $I_{-1}(C_{el})$ is calculated as in the Blatz–Ko model:

$$I_{-1} = \text{trace}(C_{el}^{-1}) = \frac{I_2(C_{el})}{I_3(C_{el})}$$

Gao proposed that the material is unconditionally stable when the parameters are bounded to $1 < n < 3$ and $0 < a$, and related these parameters under small strain to the Young's modulus and Poisson's ratio by

$$E = \frac{3^n n^2 8a}{2n+1} \text{ and } \nu = \frac{n-1}{2n+1}$$

Since $n = (1+\nu)/(1-2\nu)$ and it is bounded to $1 < n < 3$, this material is stable for an initial Poisson's ratio in the range of $0 < \nu < 2/7$.

See also the description of the Gao material model in the Solid Mechanics interface documentation.

MURNAGHAN

The Murnaghan strain energy density is used in nonlinear acoustoelasticity. Most conveniently, it is expressed in terms of the three invariants of the elastic Green–Lagrange strain tensor, $I_1(\varepsilon_{el})$, $I_2(\varepsilon_{el})$, and $I_3(\varepsilon_{el})$:

$$W_s = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2 + \frac{1}{3}(l + 2m)I_1^3 - 2mI_1I_2 + nI_3$$

Here, l, m, and n (SI unit: Pa) are the Murnaghan third-order elastic moduli, which can be found experimentally for many commonly encountered materials such as steel

and aluminum, and λ and μ are the Lamé parameters. Setting the Murnaghan moduli to zero, $l = m = n = 0$, recovers a compressible Saint-Venant–Kirchhoff material.

See also the description of the Murnaghan material model in the Solid Mechanics interface documentation.

DELFINO

Delfino proposed a simple hyperelastic material for modeling carotid arteries (Ref. 29). The strain energy density is defined by two parameters, a (SI unit: Pa) and b (dimensionless), and the first invariant of the isochoric elastic right Cauchy–Green deformation tensors $I_1(\overline{C_{\text{el}}})$:

$$W_{\text{iso}} = \frac{a}{b}\left(e^{\frac{b}{2}(\overline{I}_1 - 3)} - 1\right)$$

The parameter a plays the role of shear modulus in the small-strain limit.

The **Compressible, uncoupled** and **Nearly incompressible** formulations use the elastic volume ratio J_{el} and the bulk modulus κ to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the isochoric strain energy, and an extra variable is added to enforce the incompressibility condition $J_{\text{el}} = 1$, see Incompressible Hyperelastic Materials.

FUNG

Fung (Ref. 29) proposed one of the most popular strain energy functions to describe the deformation in soft biological tissues and arteries. For this compressible hyperelastic material, the strain energy density reads

$$W_{\text{s}} = \frac{c}{2}(e^{Q} - 1)$$

the parameter c (SI unit: Pa) scales the global stiffness, and the quadratic form Q (dimensionless) depends on the Green–Lagrange strain

$$Q = \mathbf{E}^{T} \cdot A \cdot \mathbf{E}$$

here, A is a 6-by-6 symmetric matrix (dimensionless), and $\mathbf{E}$ is a vector which contains the elements of the elastic Green–Lagrange strain tensor sorted in either standard or Voigt order, see Orthotropic and Anisotropic Materials.

For modeling the deformation in arteries, Fung proposed that the principal directions of the strain tensor coincide with the radial, circumferential, and axial directions of the artery, so the coefficient matrix A is described by nine independent parameters (Ref. 29), see Orthotropic Material to see the structure of such matrix.

The **Compressible, uncoupled** and **Nearly incompressible** formulations use the components of the isochoric elastic Green–Lagrange strain $\overline{E}_{\mathrm{el}}$ to define the isochoric strain energy density

$$W_{\mathrm{iso}} = \frac{c}{2}(e^{\bar{Q}} - 1)$$

where

$$\bar{Q} = \bar{\mathbf{E}}^{T} \cdot A \cdot \bar{\mathbf{E}}$$

Also, these formulations use the elastic volume ratio J_{el} and the bulk modulus κ to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the same isochoric strain energy, but an extra variable is added to enforce the incompressibility condition $J_{\mathrm{el}} = 1$, see Incompressible Hyperelastic Materials.

EXTENDED TUBE

The extended tube model is a micromechanics-inspired material model that considers the network constraints from molecular chains and the limited chain extensibility (Ref. 33–35). The isochoric strain energy density of the model consists of two terms

$$W_{\mathrm{iso}} = W_{\mathrm{c}} + W_{\mathrm{e}}$$

The term W_{c} represents the energy from the crosslinking of the network, and W_{e} represents the energy from confining tube constrains. These are written as

$$W_{\mathrm{c}} = \frac{G_{\mathrm{c}}}{2}\left(\frac{(1-\alpha)(\bar{I}_1 - 3)}{1-\alpha(\bar{I}_1 - 3)} + \log(1-\alpha(\bar{I}_1 - 3))\right)$$

and

$$W_c = \frac{2G_c}{\beta^2}(\bar{\lambda}_{el1}^{-\beta} + \bar{\lambda}_{el2}^{-\beta} + \bar{\lambda}_{el3}^{-\beta} - 3)$$

Here, G_c and G_e are material parameters (SI unit: Pa), $\bar{I}_1 = I_1(\bar{C}_{el})$ is the first invariant of the elastic right Cauchy–Green deformation tensor, $\bar{\lambda}_{el}$ are the isochoric principal stretches, and α and β are dimensionless coefficients.

The shear modulus in the small-strain limit depends on the parameters G_c, G_e, and α, see Equivalent Elastic Properties. Furthermore, the parameter α corresponds to the maximum finite chain extensibility, as the strain energy W_c is singular for stretch values such as $\bar{I}_1 - 3 = 1/\alpha$.

The **Compressible, uncoupled** and **Nearly incompressible** formulations use the elastic volume ratio J_{el} and the bulk modulus κ to define the volumetric strain energy density W_{vol}, see Volumetric Response and Nearly Incompressible Hyperelastic Materials.

The **Incompressible** option uses the isochoric strain energy, and it adds an extra variable to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials.

Note that the material might be unstable for certain combinations of material parameters, see Ref. 33-34 for a valid range of parameter values.

USER DEFINED

It is possible to define the strain energy density for compressible, nearly incompressible or incompressible hyperelastic materials when selecting the user defined option.

For **Compressible** hyperelastic materials, enter an expression for the elastic strain energy W_s, which can include any expressions involving the following:

- Components of C_{el}, the elastic right Cauchy–Green deformation tensor in the local material coordinate system.
- Principal invariants of C_{el}

$$I_1(C_{el}) = \text{trace}(C_{el})$$

$$I_2(C_{el}) = \frac{1}{2}(I_1^2(C_{el}) - \text{trace}(C_{el}^2))$$

$$I_3(C_{el}) = \det(C_{el})$$

The internal variables for these invariants are named `solid.I1Cel`, `solid.I2Cel`, and `solid.I3Cel`.

- Components of the elastic Green–Lagrange strain tensor ε_{el} in the local coordinate system.
- Principal elastic stretches λ_{el1}, λ_{el2}, and λ_{el3}, which are the square root of the eigenvalues of the elastic right Cauchy–Green deformation tensor C_{el}.

The internal variables for the principal elastic stretches are named `solid.stchelp1`, `solid.stchelp2`, and `solid.stchelp3`.

- Invariants of the elastic Green–Lagrange strain tensor. Since

$$\varepsilon_{el} = \frac{1}{2}(C_{el} - \mathbf{I})$$

the invariants of ε_{el} are written in terms of the invariants of C_{el}:

$$I_1(\varepsilon_{el}) = \text{trace}(\varepsilon_{el}) = \frac{1}{2}(I_1(C_{el}) - 3)$$

$$I_2(\varepsilon_{el}) = \frac{1}{2}(I_1^2(\varepsilon_{el}) - \text{trace}(\varepsilon_{el}^2)) = \frac{1}{4}(I_2(C_{el}) - 2I_1(C_{el}) + 3)$$

$$I_3(\varepsilon_{el}) = \det(\varepsilon_{el}) = \frac{1}{8}(I_3(C_{el}) - I_2(C_{el}) + I_1(C_{el}) - 1)$$

The internal variables for these invariants are named `solid.I1eel`, `solid.I2eel`, and `solid.I3eel`.

When the **Nearly incompressible** option is selected, the elastic strain energy is decoupled into the volumetric and isochoric components:

- The volumetric strain energy W_{vol}, which can be an expression involving the elastic volume ratio $J = \det(F_{el})$
- The isochoric strain energy, W_{iso}, as an expression involving the invariants of the isochoric elastic right Cauchy–Green tensor $I_1(\overline{C_{el}})$ and $I_2(\overline{C_{el}})$; the invariants of

the isochoric elastic Green–Lagrange strain $I_1(\overline{\varepsilon_{el}})$, $I_2(\overline{\varepsilon_{el}})$, and $I_3(\overline{\varepsilon_{el}})$; or the principal isochoric elastic stretches defined as $\bar{\lambda}_{eli} = \lambda_{eli}/J_{el}^{1/3}$.

When the **Nearly incompressible** option is selected, an extra variable is added to map the pressure, see The Locking Problem and Nearly Incompressible Hyperelastic Materials for details.

The internal variables for J_{el}, $I_1(\overline{C_{el}})$, and $I_2(\overline{C_{el}})$ are named `solid.Jel`, `solid.I1CIel`, and `solid.I2CIel`.

The internal variables for $I_1(\overline{\varepsilon_{el}})$, $I_2(\overline{\varepsilon_{el}})$, and $I_3(\overline{\varepsilon_{el}})$ are named `solid.I1eIel`, `solid.I2eIel`, and `solid.I3eIel`.

When the **Incompressible** option is selected, enter an expression for the isochoric elastic strain energy W_{iso}, as is done for the **Nearly incompressible** option. An extra variable is added to enforce the incompressibility condition $J_{el} = 1$, see Incompressible Hyperelastic Materials and Mixed Formulation.

The strain energy density must not contain any other expressions involving displacement or their derivatives. Examples of such expressions are components of the displacement gradient $\nabla\mathbf{u}$ and deformation gradient $F = \nabla\mathbf{u} + I$ tensors, their transpose, inversions, as well as the global material system components of C and ε. If they occur, such variables are treated as constants during symbolic differentiations.

Select the **Use elastic deformation gradient** checkbox to define the elastic strain energy density W_s, the isochoric strain energy density, W_{iso}, and the volumetric strain energy, W_{vol}, in terms of the components the elastic deformation gradient F_{el}.

Then, the second Piola–Kirchhoff stress is defined from

$$S = F^{-1}\frac{\partial W_s}{\partial F}.$$

See also the description of the User Defined material model in the Solid Mechanics interface documentation.

EQUIVALENT ELASTIC PROPERTIES

Similarly to a Linear Elastic Material, the *elasticity tensor* (or tangent stiffness) for hyperelastic materials is also defined as the fourth-order tensor

$$\mathsf{C} = 2\frac{\partial S}{\partial C} = \frac{\partial^2 W_s}{\partial C \partial C}$$

This tensor is symmetric and it possesses minor and major symmetries. Due to this, it can be represented by a symmetric 6-by-6 matrix with elements sorted as

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\ D_{12} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{13} & D_{23} & D_{33} & D_{34} & D_{35} & D_{36} \\ D_{14} & D_{24} & D_{34} & D_{44} & D_{45} & D_{46} \\ D_{15} & D_{25} & D_{35} & D_{45} & D_{55} & D_{56} \\ D_{16} & D_{26} & D_{36} & D_{46} & D_{56} & D_{66} \end{bmatrix} = \begin{bmatrix} \mathsf{C}^{1111} & \mathsf{C}^{1122} & \mathsf{C}^{1133} & \mathsf{C}^{1112} & \mathsf{C}^{1123} & \mathsf{C}^{1113} \\ \mathsf{C}^{1122} & \mathsf{C}^{2222} & \mathsf{C}^{2233} & \mathsf{C}^{2212} & \mathsf{C}^{2223} & \mathsf{C}^{2213} \\ \mathsf{C}^{1133} & \mathsf{C}^{2233} & \mathsf{C}^{3333} & \mathsf{C}^{3312} & \mathsf{C}^{3323} & \mathsf{C}^{3313} \\ \mathsf{C}^{1112} & \mathsf{C}^{2212} & \mathsf{C}^{3312} & \mathsf{C}^{1212} & \mathsf{C}^{1223} & \mathsf{C}^{1213} \\ \mathsf{C}^{1123} & \mathsf{C}^{2223} & \mathsf{C}^{3323} & \mathsf{C}^{1223} & \mathsf{C}^{2323} & \mathsf{C}^{2313} \\ \mathsf{C}^{1113} & \mathsf{C}^{2213} & \mathsf{C}^{3313} & \mathsf{C}^{1213} & \mathsf{C}^{2313} & \mathsf{C}^{1313} \end{bmatrix}$$

which is referred to as the *elasticity matrix*. The subindex ij corresponds to the local coordinate system, this is $D_{11} = \mathsf{C}^{1111} = 2\partial S_{11}/\partial C_{11}$.

The internal variables D_{11}, D_{12}, and so on are named `item.D11`, `item.D12`, and so on. Here, `item` is the name of the parent node (for instance, `solid` for a Solid Mechanics interface).

In the small-strain limit, it is often possible to relate the material parameters of hyperelastic material models to those of an equivalent isotropic linear elastic material, see Isotropic Material and Elastic Moduli. These equivalent properties are provided in Table 3-3 below, and they are accessible in COMSOL Multiphysics with the internal variables `solid.Gequ`, `solid.Kequ`, `solid.Eequ`, and `solid.nuequ`, where `solid` in the name of the physics interface. If the material model is set to **Incompressible**, the equivalent bulk modulus and Poisson's ratio are defined as `Inf`, and `0.5`, respectively.

TABLE 3-3: EQUIVALENT ELASTIC PROPERTIES OF HYPERELASTIC MATERIAL MODELS.

MATERIAL MODEL	EQUIVALENT SHEAR MODULUS	EQUIVALENT BULK MODULUS
Neo-Hookean	μ	$\lambda + 2\mu/3$
Saint-Venant–Kirchhoff	μ	$\lambda + 2\mu/3$
Mooney–Rivlin	$2(C_{10} + C_{01})$	κ

TABLE 3-3: EQUIVALENT ELASTIC PROPERTIES OF HYPERELASTIC MATERIAL MODELS.

MATERIAL MODEL	EQUIVALENT SHEAR MODULUS	EQUIVALENT BULK MODULUS
Yeoh	$2c_1$	κ
Ogden	$\frac{1}{2}\sum_p \mu_p \alpha_p$	κ
Storakers	$\sum_k \mu_k$	$\sum_k 2\mu_k\left(\beta_k + \frac{1}{3}\right)$
Varga	$\frac{1}{2}(c_1 + c_2)$	κ
Arruda–Boyce	$2\mu_0 \sum_{p=1} p c_p \left(\frac{3}{N}\right)^{p-1}$	κ
Gent	μ	κ
van der Waals	μ	κ
Blatz–Ko	μ	$2\mu\left(\beta + \frac{1}{3}\right)$
Gao	$4an3^{n-1}$	$8an^2 3^{n-2}$
Murnaghan	μ	$\lambda + 2\mu/3$
Delfino	a	κ
Fung	Voigt estimate	Voigt estimate
Extended Tube	$(1-2\alpha)G_c + G_e$	κ
User defined	From user input	From user input

MULLINS EFFECT

Some nonlinear effects observed in rubbers, such as hysteresis in stress–stretch curves, residual strains, and stress softening effects, are not accounted in the formulation of common hyperelastic materials. The Mullins effect (Ref. 43-45) describes the stress-softening phenomenon observed under cyclic loading in elastomers and biological materials.

Ogden and Roxburgh (Ref. 46) used an additional state variable to model the Mullins effect. The state variable η is introduced to memorize the microstructural damage on reinforced rubber after repeated loading-unloading cycles. The modified isochoric strain energy density reads

$$\overline{W}_{\text{iso}} = \eta W_{\text{iso}} + \phi(\eta)$$

here, W_{iso} is the isochoric strain energy of the undamaged material, and $\phi(\eta)$ is referred as the *damage function*. The choice of the damage function $\phi(\eta)$ is completely arbitrary as long as some constrains are fulfilled. The authors (Ref. 46) proposed a state function based on the *error function*, which defined how the state variable η varies as a function of the isochoric strain energy

$$\eta = 1 - \frac{1}{r}\text{erf}\left(\frac{W_{max} - W_{iso}}{m}\right)$$

here, erf(.) is the error function, r and m are positive parameters, and W_{max} is the maximum attained value of the isochoric strain energy density on the loading path.

The associated microstructural damage is computed from

$$d = 1 - \eta = \frac{1}{r}\text{erf}\left(\frac{W_{max} - W_{iso}}{m}\right)$$

and the isochoric strain energy modified by the Mullins effect reads

$$\overline{W}_{iso} = (1 - d)W_{iso}$$

Over the years others authors (Ref. 47-48) have proposed different flavors of Ogden–Roxburgh model, the version implemented in COMSOL Multiphysics uses by default the *hyperbolic tangent function*, and a parameter to define the maximum allowed damage $d_\infty = 1/r$. The microstructural damage is then computed from

$$d = d_\infty \tanh\left(\frac{W_{max} - W_{iso}}{W_{sat} + \alpha W_{max}}\right)$$

Here, d_∞, α, and W_{sat} are positive parameters. It is also possible to use the *error function* as in the original formulation (Ref. 46), in which case the microstructural damage is computed from

$$d = d_\infty \text{erf}\left(\frac{W_{max} - W_{iso}}{W_{sat} + \alpha W_{max}}\right)$$

Miehe (Ref. 49) proposed an exponential expression for the damage variable in order to model the Mullins effect

$$d = d_\infty (1 - e^{-W_{max}/W_{sat}})$$

where W_{sat} and d_∞ are positive parameters.

Distributed Fiber Models

Composite materials are made out of an isotropic matrix reinforced with fibers. In these reinforced materials, the fibers act as the main load carrier, while the surrounding matrix supports the fibers and transfers loads to them. These materials are either manufactured or have a natural origin such as biological tissue, leafs, or wood.

In COMSOL Multiphysics it is possible to add fibers to Linear Elastic or Nonlinear Elastic materials, as well as to any of the built-in Hyperelastic material models. The fibers are arranged in a specified orientation inside the matrix material. By subsequently adding **Fiber** nodes as needed, it is possible to include fibers in many directions, thus recreating complex anisotropic structures.

FIBERS FOR LINEAR AND NONLINEAR ELASTIC MATERIALS

For a matrix material described by a Linear Elastic or Nonlinear Elastic materials, a **Fiber** described by a **Linear Elastic** material adds a contribution to the stress tensor of the type

$$\sigma_{\text{fib}} = v_{\text{fib}}(E_{\text{fib}} - E)\varepsilon_{\text{a}}\mathbf{a} \otimes \mathbf{a}$$

Here, $E_{\text{fib}} - E$ is the difference between the stiffness of the isotropic matrix E and the fibers' stiffness E_{fib}, v_{fib} is the fiber volume fraction, $\mathbf{a}$ is the direction vector for the fiber orientation in the undeformed geometry, $\varepsilon_{\text{a}} = \mathbf{a} \cdot \varepsilon_{\text{el}} \cdot \mathbf{a}$ is the elastic strain in the fiber direction $\mathbf{a}$, and $\otimes$ is the outer product of two vectors, $(\mathbf{a}\otimes\mathbf{a})_{ij} = a_i a_j$.

For a Linear Elastic Material, Hooke's law (Equation 3-21) is augmented as

$$\sigma = \sigma_{\text{ex}} + \mathsf{C} : \varepsilon_{\text{el}} + \sigma_{\text{fib}}$$

For Nonlinear Elastic Materials, the same contribution to the stress tensor is added, although the stress–strain relation in the isotropic matrix could be nonlinear.

When the fiber stiffness is described by a **Uniaxial stress function**, the added stiffness reads

$$\sigma_{\text{fib}} = v_{\text{fib}}(\sigma_{\text{a}}(\varepsilon_{\text{a}}) - E\varepsilon_{\text{a}})\mathbf{a} \otimes \mathbf{a}$$

where $\sigma_{\text{a}}(\varepsilon_{\text{a}})$ is a function of the elastic strain in the fiber direction, $\varepsilon_{\text{a}} = \mathbf{a} \cdot \varepsilon_{\text{el}} \cdot \mathbf{a}$. A linear elastic fiber can be described by

$$\sigma_{\text{a}}(\varepsilon_{\text{a}}) = E_{\text{fib}}\varepsilon_{\text{a}}$$

but in general, $\sigma_{\text{a}}(\varepsilon_{\text{a}})$ could be a nonlinear expression of the fiber strain ε_{a}.

FIBERS FOR HYPERELASTIC MATERIALS

For Hyperelastic Materials, the **Holzapfel–Gasser–Ogden** model is available beside the **Linear elastic** and **Uniaxial data** models for fibers. In this case, the fibers are represented by additional contributions to the strain energy density (Ref. 29 and Ref. 30).

The Holzapfel–Gasser–Ogden constitutive model captures the anisotropic mechanical response observed in arteries. The model combines a nearly incompressible material for the isotropic matrix, with additional terms for two fiber families

$$W_{\text{ani}} = W_{\text{iso}} + W_{\text{fib},1} + W_{\text{fib},2}$$

Here, W_{iso} is the strain energy density of the isotropic hyperelastic material. In Ref. 29 W_{iso} is represented by a nearly incompressible Neo-Hookean material; in COMSOL Multiphysics it is possible to select any of the built-in hyperelastic material models.

The second and third terms, $W_{\text{fib},1}$ and $W_{\text{fib},2}$ are anisotropic additions that describe the mechanical contribution of collagen fiber networks. These terms contribute with a strain energy density of the type

$$W_{\text{fib}} = \frac{k_1}{2k_2}(e^Q - 1) \text{ where } Q = k_2(k_3(\overline{I}_1 - 3) + (1 - 3k_3)(\overline{I}_a - 1))^2$$

where the parameter k_1 represents the fiber stiffness (SI unit: Pa), k_2 is a dimensionless tuning parameter, and k_3 is the fiber dispersion (dimensionless). A value of $k_3 = 0$ recovers the formulation in Ref. 29, where all the fibers are perfectly oriented in the $\mathbf{a}$ direction. Setting $k_3 = 1/3$ means that the fibers are completely dispersed, thus recovering a similar formulation as given in the Delfino material model.

The formulation in Ref. 29 and Ref. 30 is based on the Isochoric Elastic Deformation, where the invariants are defined as

$$\overline{I}_1 = \text{trace}(\overline{C}_{\text{el}})$$

$$\overline{I}_a = \mathbf{a} \cdot \overline{C}_{\text{el}} \cdot \mathbf{a}$$

The invariant $\overline{I}_a = \overline{\lambda}_{\text{a}}^2$ represents the squared value of the isochoric elastic stretch in the fiber direction $\mathbf{a}$. A common modeling assumption for biological tissues is that fibers cannot sustain compression, so the fiber stiffness is added only for tensile stretches such as $\overline{\lambda}_{\text{a}} > 1$.

In general, the anisotropic strain energy density for a hyperelastic material containing N fiber families reads

$$W_{\text{ani}} = W_{\text{iso}} + \sum_{m=1}^{N} W_{\text{fib},\,m}$$

The second Piola–Kirchhoff stress tensor is then computed from

$$S = S_{\text{iso}} + \sum_{m=1}^{N} S_{\text{fib},\,m}$$

where the stress contribution from each fiber family is given by

$$S_{\text{fib},\,m} = 2\frac{\partial W_{\text{fib},\,m}}{\partial C}$$

COMPRESSIBLE FIBERS

The hyperelastic formulation in the Holzapfel–Gasser–Ogden constitutive model (Ref. 29 and Ref. 30) based on the isochoric elastic Cauchy–Green tensor can be problematic when the matrix material is made of a compressible or nearly incompressible material (Ref. 31 and Ref. 32).

The solution to this problem (Ref. 32) is to relax the isochoric restriction for the fiber deformation, and use the total strain invariant in the fiber direction instead. The contribution to the strain energy density then reads

$$W_{\text{fib}} = \frac{k_1}{2k_2}(e^{Q} - 1) \text{ where } Q = k_2(k_3(I_1 - 3) + (1 - 3k_3)(I_a - 1))^2$$

where the invariants are defined as

$$I_1 = \text{trace}(C_{\text{el}})$$

$$I_a = \mathbf{a} \cdot C_{\text{el}} \cdot \mathbf{a}$$

USER DEFINED

The anisotropic strain energy density for the fiber family W_{fib} can also be specified as a user-defined expression of the strain invariants I_a, I_b, I_c, I_{ab}, I_{ac}, and I_{bc}, where

$$I_b = \mathbf{b} \cdot C_{\text{el}} \cdot \mathbf{b}, I_c = \mathbf{c} \cdot C_{\text{el}} \cdot \mathbf{c}$$

$$I_{ab} = \mathbf{a} \cdot C_{\text{el}} \cdot \mathbf{b}, I_{ac} = \mathbf{a} \cdot C_{\text{el}} \cdot \mathbf{c}, \text{ and } I_{bc} = \mathbf{b} \cdot C_{\text{el}} \cdot \mathbf{c}$$

or the isochoric version of these invariants $\overline{I_a}$, $\overline{I_b}$, $\overline{I_c}$, $\overline{I_{ab}}$, $\overline{I_{ac}}$, and $\overline{I_{bc}}$, where

$$\overline{I_b} = \mathbf{b} \cdot \overline{C_{\text{el}}} \cdot \mathbf{b}, \overline{I_c} = \mathbf{c} \cdot \overline{C_{\text{el}}} \cdot \mathbf{c}$$

$$\overline{I_{ab}} = \mathbf{a} \cdot \overline{C_{\text{el}}} \cdot \mathbf{b}, \overline{I_{ac}} = \mathbf{a} \cdot \overline{C_{\text{el}}} \cdot \mathbf{c}, \text{ and } \overline{I_{bc}} = \mathbf{b} \cdot \overline{C_{\text{el}}} \cdot \mathbf{c}$$

here, **b** and **c** are orthogonal directions to the main fiber direction **a**.

The internal variables $I_a, I_b, I_c, I_{ab}, I_{ac}, I_{bc}, \overline{I_a}, \overline{I_b}, \overline{I_c}, \overline{I_{ab}}, \overline{I_{ac}}$, and $\overline{I_{bc}}$ are named *item*`.IaCe`, *item*`.IbCe`, *item*`.IcCe`, *item*`.IabCe`, *item*`.IacCe`, *item*`.IbcCe`, *item*`.IaCIe`, *item*`.IbCIe`, *item*`.IcCIe`, *item*`.IabCIe`, *item*`.IacCIe`, and *item*`.IbcCIe`, respectively. Here, *item* is the name of the *Fiber* node (for instance, `solid.hmm1.fib1`).

THERMAL EXPANSION IN FIBERS

Not only the stiffness in the fibers might differ by orders of magnitude to the matrix stiffness, but also the thermal properties, so it is possible to model thermal expansion in fibers independently from the thermal expansion in the surrounding material. The two basic assumptions are that the volume occupied by the fibers is small as compared to the base material and that thermal expansion (or contraction) occurs in the fiber direction only (there is zero transverse thermal expansion in the fibers).

For Linear Elastic and Nonlinear Elastic materials, fibers described by a Linear Elastic material add a contribution to the stress tensor of the type

$$\sigma_{\text{fib}} = v_{\text{fib}}(E_{\text{fib}} - E)\varepsilon_{\text{a}}\mathbf{a} \otimes \mathbf{a}$$

where $E_{\text{fib}} - E$ is the difference between the stiffness of the isotropic matrix E and the fibers' stiffness E_{fib}, v_{fib} is the fiber volume fraction, **a** is the direction vector for the fiber orientation in the undeformed geometry, and $\varepsilon_{\text{a}} = \mathbf{a} \cdot \varepsilon_{\text{el}} \cdot \mathbf{a}$ is the elastic strain in the fiber direction **a**.

When considering thermal expansion in the fibers, the stress reads

$$\sigma_{\text{fib}} = v_{\text{fib}}(E_{\text{fib}}(\varepsilon_{\text{a}} - \varepsilon_{\text{th,fib}}) - E\varepsilon_{\text{a}})\mathbf{a} \otimes \mathbf{a}$$

where $\varepsilon_{\text{th,fib}}$ is the thermal strain in the fibers. When the fiber stiffness is described by a **Uniaxial stress function**, the added stiffness reads

$$\sigma_{\text{fib}} = v_{\text{fib}}(\sigma_{\text{a}}(\varepsilon_{\text{a}}) - E\varepsilon_{\text{a}})\mathbf{a} \otimes \mathbf{a}$$

where $\sigma_a(\varepsilon_a)$ is a function of the elastic strain in the fiber direction, where

$$\varepsilon_a = \mathbf{a} \cdot \varepsilon \cdot \mathbf{a} - \varepsilon_{th,fib} .$$

For a Linear Elastic Material, Hooke's law (Equation 3-21) is then augmented as

$$\sigma = \sigma_{ex} + \mathsf{C} : \varepsilon_{el} + \sigma_{fib}$$

For Nonlinear Elastic Materials, the same contribution to the stress tensor is added, although the stress–strain relation in the isotropic matrix could be nonlinear.

For Hyperelastic Materials containing one fiber family, the anisotropic strain energy density reads:

$$W_{ani} = W_{iso} + W_{fib}$$

The second Piola–Kirchhoff stress tensor is then computed from

$$S = S_{iso} + 2\frac{\partial W_{fib}}{\partial C}$$

When **Thermal Expansion** is added to fibers described by the **Holzapfel–Gasser–Ogden** material model, the fiber contribution W_{fib} to the strain energy density is computed from the invariants of the elastic Cauchy–Green tensor:

$$C_{el} = F_{th}^{-T} C F_{th}^{-1}$$

where the inelastic deformation gradient of the fibers is

$$F_{th} = I + \varepsilon_{th,fib} \mathbf{a} \otimes \mathbf{a} .$$

Thermal Strain in Fibers

The thermal strain in the fibers, $\varepsilon_{th,fib}$, can be specified by different means. When the secant coefficient of thermal expansion is used, it reads

$$\varepsilon_{th,fib} = \alpha_s (T_{fib} - T_{ref})$$

Here, the secant coefficient of thermal expansion α_s can be temperature-dependent. The reference temperature T_{ref} is the temperature at which there are no thermal strains in the fibers, and T_{fib} is the current fiber temperature.

When the tangent coefficient of thermal expansion is used, α_t, the thermal strain is given by

$$\varepsilon_{\text{th,fib}} = \exp\left(\int_{T_{\text{ref}}}^{T_{\text{fib}}} \alpha_t(\tau)d\tau\right) - 1$$

It is also possible to explicitly enter the thermal strain, dL, as a function of the fiber's temperature

$$\varepsilon_{\text{th,fib}} = dL(T_{\text{fib}})$$

The coefficient of the thermal expansion or the thermal strain are considered in the fiber direction **a**, since thermal expansion is assumed to occur in the fiber direction only.

	See also the description of the Fiber node in the *Solid Mechanics* chapter.

	An example of an isotropic hyperelastic material reinforced with fibers is shown in *Arterial Wall Mechanics*: Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/arterial_wall_mechanics**

Elastoplastic Materials

Many materials exhibit a distinct elastic regime where deformations are recoverable and path-independent. This elastic behavior can be modeled using a Linear Elastic Material, Nonlinear Elastic Materials, or Hyperelastic Materials models. In elastoplasticity, the elastic regime is constrained by the *yield limit* in stress space. When stresses exceed this limit, permanent inelastic strains develop, which, unlike elastic strains, remain even after the stresses are removed.

For elastoplastic materials, the emphasis is primarily on models that capture inelastic strains developing independently of loading rate, known as *rate-independent* plasticity models. This class of models effectively describes many engineering materials, including homogeneous or porous metals at low temperatures, as well as soils.

In contrast, *rate-dependent* elastoplastic material models, such as creep and viscoplasticity material models, are commonly applied in scenarios involving high temperatures, where a clear yield limit is difficult to define, or under conditions of high loading rates. For more details on these models, refer to Creep and Viscoplasticity.

In this section:

- The Plasticity Problem
- Metal Plasticity
- Pressure-Dependent and Soil Plasticity
- Isotropic Hardening
- Kinematic Hardening
- Mixed Hardening

THE PLASTICITY PROBLEM

A general elastoplastic constitutive model using the flow theory of plasticity includes the following basic components:

- A decomposition of strains as described in Inelastic Strain Contributions
- An elastic law
- A yield criterion
- A plastic flow rule that defines the evolution of plastic strains
- A hardening law that defines the evolution of the yield limit
- A hardening rule that defines the evolution of hardening variables

The elastoplastic constitutive model is a dissipative model that can be formulated within the framework of thermodynamics using internal variables. In plasticity, the internal variables typically include the plastic strain tensor ε_{pl} and a set of hardening variables α.

Yield Criterion

Plastic deformation occurs when the stresses reach a critical value. In a three-dimensional setting, this may be expressed by a scalar *yield function* F_y.

For a general plasticity model, it can be expressed as

$$F_y(\sigma, A) = 0 \tag{3-53}$$

where σ is the total stress tensor and A is a set of hardening forces. In case of a geometrically nonlinear case, σ is typically identified as the Cauchy stress tensor. The hardening forces depend on the hardening variables α. In many cases there is only a single hardening force, the scalar yield stress σ_{ys}. The yield function then reads

$$F_y(\sigma, \sigma_{ys}) = 0 \tag{3-54}$$

The *elastic domain* is defined as the set of stresses for which the yield function F_y is smaller than zero, that is

$$\Omega_e = \{\sigma | F_y(\sigma, \sigma_{ys}) < 0\}$$

For stresses in Ω_e no plastic deformation occurs. However, for stresses on the boundary of the elastic domain, where $F_y = 0$, plastic deformation may occur. This boundary is called the *yield surface*, which is represented by a hypersurface in stress space. It can be defined as

$$\Upsilon = \{\sigma | F_y(\sigma, \sigma_{ys}) = 0\}$$

Plastic Flow Rule

The *flow rule* defines the evolution of the plastic deformation. The flow rule is defined in terms of a *plastic potential* Q_p, which is given on the same general form as the yield function

$$Q_p = Q_p(\sigma, A)$$

The gradient of the plastic potential Q_p with respect to the stress tensor σ gives the direction of plastic flow as

$$G_{pl} = \frac{\partial Q_p}{\partial \sigma}$$

For *associated plastic flow*, the plastic potential equals the yield function, that is

$$Q_p = F_y$$

When the plastic potential is different from the yield function, the flow rule is said to be *nonassociated*.

Additive Plastic Flow When an additive decomposition of strains is used, COMSOL Multiphysics characterizes plastic deformation through the plastic strain tensor ε_{pl}. The flow rule then describes how ε_{pl} evolves. This equation is postulated as

$$\dot{\varepsilon}_{pl} = \lambda G_{pl} = \lambda \frac{\partial Q_p}{\partial \sigma} \tag{3-55}$$

Here, λ is the *plastic multiplier*, which is a nonnegative variable that satisfies the complementarity condition $\lambda F_y = 0$. The latter condition states that plastic flow may only occur when the stress is *on* the yield surface.

When additive plasticity is used and the **Include geometric nonlinearity** checkbox is selected in the study **Settings** window, the Cauchy stress tensor is used to evaluate the yield function and plastic potential. Plastic flow is, however, assumed to be in the direction of the second Piola–Kirchhoff stress tensor, and the computed plastic strain tensor can be interpreted as a Green–Lagrange strain.

Multiplicative Plastic Flow When a multiplicative decomposition of deformation is used, COMSOL Multiphysics characterizes plastic deformation through the plastic right Cauchy–Green tensor C_p, or more precisely its inverse C_p^{-1}. However, the internal variable is the inverse of the (symmetric) plastic stretch tensor, U_p^{-1}, such that $C_p^{-1} = U_p^{-2}$. An exception is when kinematic hardening is included. Then plastic deformation is characterized by the inverse of the (nonsymmetric) plastic deformation tensor, F_p^{-1}. See Kinematic Hardening for details.

In a multiplicative formulation of plasticity (Ref. 93, Ref. 94, and Ref. 95), the plastic flow rule can be written as the *Lie derivative* of the elastic left Cauchy–Green deformation tensor B_{el}:

$$-\frac{1}{2}\mathcal{L}(B_{el}) = \lambda G_{pl} B_{el} = \lambda \frac{\partial Q_p}{\partial \sigma} B_{el} \tag{3-56}$$

As for the Additive Plastic Flow, λ represents the plastic multiplier, which is a nonnegative variable that satisfies the complementarity condition $\lambda F_y = 0$. The latter condition states that plastic flow may only occur when the stress is on the yield surface.

The Lie derivative of B_{el} is then expressed in terms of the rate of the plastic right Cauchy–Green tensor

$$\mathcal{L}(B_{el}) = F \dot{C}_p^{-1} F^T \tag{3-57}$$

By using Equation 3-56 and Equation 3-57, the plastic flow rule is written as (Ref. 94)

$$-\frac{1}{2} F \dot{\overline{C_p^{-1}}} F^T = \lambda G_{pl} B_{el}$$

Finally, the elastic left Cauchy–Green tensor is expressed in terms of the deformation gradient $\boldsymbol{F}$ and $\boldsymbol{C}_\mathrm{p}$, so $\boldsymbol{B}_\mathrm{el} = \boldsymbol{F}\boldsymbol{C}_\mathrm{p}^{-1}\boldsymbol{F}^T$. The flow rule then reads

$$\dot{\overline{\boldsymbol{C}_\mathrm{p}^{-1}}} = -2\lambda \boldsymbol{F}^{-1}\boldsymbol{G}_\mathrm{pl}\boldsymbol{F}\boldsymbol{C}_\mathrm{p}^{-1} \tag{3-58}$$

For results evaluation, the plastic Green–Lagrange strain tensor is defined as

$$\boldsymbol{\varepsilon}_\mathrm{pl} = \frac{1}{2}(\boldsymbol{C}_\mathrm{p} - \boldsymbol{I})$$

Hardening Rule

The evolution of the hardening variables is described by the *hardening rule*. Similarly to the flow rule, the hardening rule for a set of hardening variables α is postulated as

$$\dot{\alpha} = \lambda H_\mathrm{p} \tag{3-59}$$

Here, $H_\mathrm{p}(\sigma, A)$ is a general hardening modulus that describes how the hardening variables evolve with plastic deformation.

In principle, any number of hardening variables may be considered and different plasticity models in COMSOL Multiphysics may add one or several such variables. However, the most common approach is to use a single hardening variable, typically the *equivalent plastic strain* ε_pe. The general hardening law in Equation 3-59 then simplifies to

$$\dot{\varepsilon}_\mathrm{pe} = \lambda h_\mathrm{p} \tag{3-60}$$

The equivalent plastic strain ε_pe is then used to define how the yield stress σ_ys evolves during plastic flow, that is, σ_ys is the thermodynamic force conjugate to ε_pe. See Isotropic Hardening for more details.

The *hardening modulus* h_p may take different forms, and it can be specified in a user defined expression.

A common assumption follows from the concept of *strain hardening*, where the definition of a von Mises equivalent strain is used

$$h_\mathrm{p} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_\mathrm{pl} : \dot{\boldsymbol{\varepsilon}}_\mathrm{pl}}$$

Another approach is to derive the hardening modulus h_p from a potential, analogous to how plastic flow is defined by taking the gradient with respect to its conjugate hardening force. Here, we assume an associated hardening law, such that

$$h_p = -\frac{\partial F_y}{\partial \sigma_{ys}}$$

Other variables are added when Kinematic Hardening is combined with a plasticity model.

The equivalent plastic strain variable is computed as the *true* equivalent plastic strain (also called *Hencky* or *logarithmic* plastic strain) when the Multiplicative Plastic Flow is used.

Loading and Unloading Conditions

The mathematical definition of the general plasticity problem is completed by specifying a set of loading and unloading conditions. These conditions are often referred to as the *Kuhn–Tucker conditions*,

$$\lambda \geq 0\,,\ F_y \leq 0 \ \text{ and } \ \lambda F_y = 0$$

As a result of the second and third conditions, it is concluded that during plastic flow, when $\lambda > 0$, it is also required that

$$F_y = 0 \tag{3-61}$$

In other words, it is required that stresses remain on the yield surface. Equation 3-61 is used to determine the value of the plastic multiplier λ during plastic flow.

Numerical Integration Algorithm

COMSOL Multiphysics uses the standard predictor-corrector algorithm to solve the general plasticity problem. The algorithm is here described schematically assuming an additive decomposition of strains.

The general plasticity problem is defined by Equation 3-53, Equation 3-59, and Equation 3-61, with internal variables ε_{pl}, α, and λ.

The first step is to apply a backward Euler scheme to discretize the evolution equation to arrive at the following set of algebraic equations

$$L(q^{n+1}, U) = \begin{bmatrix} \varepsilon_{pl}^{n+1} - \varepsilon_{pl}^{n} - \Lambda G_{pl}^{n+1} \\ \alpha^{n+1} - \alpha^{n} - \Lambda H_{pl}^{n+1} \\ F_y^{n+1} \end{bmatrix} = 0$$

Here, the superscript $n+1$ indicates a variable evaluated at the current parameter or time step, and a superscript n denotes a variable with values form the previously converged step.

Also, U represents other dependent variables in the model, such as displacements, that are held fixed within the plastic algorithm, the vector q

$$q^{n+1} = \begin{bmatrix} \varepsilon_{pl}^{n+1} \\ \alpha^{n+1} \\ \Lambda \end{bmatrix}$$

is a collection of all plastic variables, and Λ is the incremental plastic multiplier.

For every Gauss point in a domain with an elastoplastic material model, the following algorithm is applied to find the current value of the plastic variables q^{n+1}:

1. Predictor step. In the predictor step the assumption that the stress is inside the elastic domain is evaluated.
   a. Initialize the plastic variables with values from the previously converged step and $\Lambda = 0$, that is

$$q^{n+1} = \begin{bmatrix} \varepsilon_{pl}^{n} \\ \alpha^{n} \\ 0 \end{bmatrix}$$

   b. Check if the stress is within the elastic domain or on the yield surface by evaluating

$$F_y[\sigma(\varepsilon_{pl}^{n+1}), A(\alpha^{n+1})]$$

   c. If $F_y \leq 0$, the stress is elastic and no more steps are necessary. Otherwise the predictor solution is not admissible, and the algorithm continues to the corrector step.

2. Corrector step. In the corrector step COMSOL Multiphysics attempts to minimize the residual of the equation $L = 0$ to find the current values of plastic variables q^{n+1}. This is done by applying Newton's method:
   a. Initialize the iteration counter $k = 0$ and set ${}^0q = q^{n+1}$
   b. Compute the residual ${}^kL({}^kq, U)$
   c. Compute and invert the local Jacobian

$$J = \frac{\partial}{\partial q} L({}^{k}q, U)$$

d Update the plastic variables ${}^{k+1}q = {}^{k}q + \delta q = {}^{k}q - (J^{-1}){}^{k}L$

e Check the convergence by evaluating $|\delta\Lambda| < \varepsilon_{\text{rel}}|{}^{k+1}\Lambda|$. Here ε_{rel} is the relative tolerance. If converged, exit the algorithm and set $q^{n+1} = {}^{k+1}q$.

f Update the iteration counter k = k+1. If $k < k_{\max}$, go to step **b**, otherwise exit the algorithm with an error.

3 Compute the Jacobian of the plastic variables q^{n+1} with respect to other dependent variables in the model, U, if requested by the global solver.

There are additional steps in the algorithm, to ensure that the converged solution fulfills $\Lambda > 0$.

It is also possible to specify the maximum number of iterations and the relative tolerance used to solve the plastic algorithm. In the **Advanced** section of the feature, enter the following settings:

- **Maximum number of local iterations** to specify $k_{\max}$. The default value is 25 iterations.
- **Relative tolerance** to specify ε_{rel}. The default value is 10^{-6}.

The standard Newton method may not provide a robust solution for highly nonlinear plasticity models. For such models the algorithm can be enhanced by *line search iterations* in the corrector step (during step **d** above). This is done automatically for some plasticity models.

In most plasticity features it is also possible to force the use of line search iterations by setting the **Local method** to **Backward Euler, damped**. This setting is available in the **Advanced** section. When selected, it is possible to specify the **Maximum number of line search iterations**. A common value is 4. It is important to note that an excessive number of line search iterations can cause the algorithm to stall.

Numerical Integration for Multiplicative Decomposition

When using the multiplicative decomposition, the inverse of the plastic stretch tensor U_{p}^{-1} is the internal variable used to characterize plastic deformation. In general COMSOL Multiphysics here also applies the so-called exponential mapping technique to solve the plastic flow rule in Equation 3-58 as

$$(C_{\text{p}}^{-1})^{n+1} = F^{-1}\exp(-2\Lambda G_{\text{pl}}^{n+1})F(C_{\text{p}}^{-1})^{n}$$

where $(C_p^{-1})^{n+1} = (U_p^{-1})^{n+1}(U_p^{-1})^{n+1}$ and $(C_p^{-1})^n = (U_p^{-1})^n(U_p^{-1})^n$.

The matrix exponential in the discrete flow rule is computed by the spectral decomposition method.

The purpose of the exponential mapping is to ensure volume preserving plasticity of the discretized flow rule. For plasticity models that are not volume preserving, for example soil plasticity, the backward Euler discretization is used.

Once the flow rule have been discretized, the numerical solution follows identical steps as described in the Numerical Integration Algorithm section.

Multisurface Plasticity

For many engineering materials the yield surface is not sufficiently described by a single yield function, but rather requires a combination of several yield functions. Such models are referred to as multisurface plasticity models.

The singe yield function in Equation 3-53 is then extended to a set of n yield functions such as

$$F_y^i(\sigma, A) = 0, i = 1, \ldots, n$$

Similarly, the combined yield surface now reads

$$\Upsilon = \{\sigma | (F_y^i(\sigma, A) = 0, i = 1, \ldots, n)\}$$

In the general case, the combined yield surface resulting from the set of yield functions is not smooth, and in general, the same applies to the combined plastic potential. Hence, n plastic flow rules are required with corresponding plastic multipliers λ^i.

COMSOL Multiphysics implements multisurface plasticity models by converting the set of yield functions into a single smooth and C^2 differentiable function. This is done by using the procedure outlined in Ref. 118. The same procedure is applied for combining a set of n plastic potentials.

A combined yield function can be defined as

$$F_y = \max(F_y^1, \ldots, F_y^n) = 0$$

however, using the maximum operator leads to a nonsmooth function, making it challenging to obtain a robust numerical algorithm.

To address this, an iterative approach is employed, where n yield functions are combined into a single smooth function F_{ys}^{n-1}.

The procedure follows as

$$F_{ys}^{1} = \max(F_{y}^{1}, F_{y}^{2}) + s(F_{y}^{1} - F_{y}^{2})$$
$$F_{ys}^{2} = \max(F_{ys}^{1}, F_{y}^{3}) + s(F_{ys}^{1} - F_{y}^{3})$$
$$\ldots$$
$$F_{ys}^{n-1} = \max(F_{ys}^{n-2}, F_{y}^{n}) + s(F_{ys}^{n-2} - F_{y}^{n})$$

Here, $s(x)$ is a smooth function defined as

$$s(x) = \begin{cases} 0 & \text{if } |x| \geq a \\ \frac{1}{2}(a - |x|) + g(x) & \text{if } |x| < a \end{cases}$$

where a is a smoothing parameter, and $g(x)$ is a function that fulfills certain requirements among them being C^2 differentiable. A simple function that meets these requirements is the trigonometric function (Ref. 118)

$$g(x) = -\frac{a}{\pi}\cos\frac{\pi x}{2a}$$

The smoothing parameter a is then related to the offset of the smooth function F_{ys}^{n-1} from the nonsmooth function F_y at the intersections of two or more functions F_{ys}^{k}.

Depending on the plasticity model, COMSOL Multiphysics may apply this smoothing technique to both the yield function and the plastic potential, or to the plastic potential only.

Once the smooth multisurface plasticity model is defined, all other considerations, including the numerical solution, follows in the same manner as for a single surface plasticity model.

Energy Dissipation

Plasticity is an inelastic process and therefore it dissipates energy. The dissipated energy per unit volume can be calculated from an evolution equation similar to the plastic flow rule

$$\dot{W}_p = \sigma{:}\dot{\varepsilon}_p = \lambda\sigma{:}G_{pl}$$

As done for the plastic flow rule, a backward Euler scheme is used to discretize the evolution equation to arrive at the following discrete equation

$$W_p^{n+1} - W_p^n - \Lambda \sigma^{n+1} : G_{pl}^{n+1} = 0$$

from which the plastic dissipation density at the current parameter or time step, W_p^{n+1}, is computed. Here, Λ is the incremental plastic multiplier defined by the Numerical Integration Algorithm.

The total energy dissipated by plasticity in a given volume can be calculated by a volume integration of the plastic dissipation density W_p.

When the **Calculate dissipated energy** checkbox is selected, the plastic dissipation density is available under the variable `solid.Wp` and the total plastic dissipation under the variable `solid.Wp_tot`.

Nonlocal Plasticity

The nonlocal plasticity model implemented in COMSOL Multiphysics is based on the *implicit gradient* method suggested in Ref. 116, and it incorporates some of the generalizations introduced by the micromorphic theory (Ref. 117).

The theory assumes that the free energy of the elastoplastic material depends on both the macroscopic displacement u and other macrostate variables, such as the equivalent plastic strain ε_{pe}, and also on some microstate variable.

When the microstate variable is identified as a nonlocal version of the equivalent plastic strain, $\varepsilon_{pe,nl}$, the free energy potential is written as

$$\rho\psi(u, \nabla u, \varepsilon_p, \varepsilon_{pe}, \varepsilon_{pe,nl}, \nabla\varepsilon_{pe,nl}) = W_s(u, \nabla u, \varepsilon_p) + \frac{1}{2}E_{iso}(\varepsilon_{pe})^2 + \frac{1}{2}H_{nl}(\varepsilon_{pe} - \varepsilon_{pe,nl})^2 + \frac{1}{2}H_{nl}(l_{int})^2\nabla\varepsilon_{pe,nl} \cdot \nabla\varepsilon_{pe,nl} \quad (3\text{-}62)$$

where W_s is the elastic free energy, H_{nl} is a model parameter that determines the strength of the coupling between the macro and micro states, and l_{int} is the length scale that determines the influence radius of the interaction. The length scale l_{int} should ideally be larger than the largest mesh element size to ensure an adequate spatial averaging.

For simplicity, Equation 3-97 assumes linear isotropic hardening, but the same concept is applicable for more general hardening laws, and it can also be extended to kinematic hardening.

Taking the variation of Equation 3-97 with respect to the state variables ε_{pe}, $\varepsilon_{pe,nl}$, and $\nabla\varepsilon_{pe,nl}$ leads to the following set of equations related to the plasticity model:

$$\varepsilon_{pe,nl} - l_{int}^2 \nabla^2 \varepsilon_{pe,nl} = \varepsilon_{pe} \quad \text{in } \Omega$$
$$\nabla \varepsilon_{pe,nl} \cdot n = 0 \quad \text{on} \partial\Omega$$
$$\sigma_{ys}(\varepsilon_{pe}, \varepsilon_{pe,nl}) = \sigma_{ys0} + E_{iso}\varepsilon_{pe} + H_{nl}(\varepsilon_{pe} - \varepsilon_{pe,nl})$$

here, Ω represents the domain of the elastoplastic problem.

The variation of Equation 3-97 with respect to the remaining state variables leads to the standard elasticity equations. In the nonlocal (or regularized) plasticity problem, $\varepsilon_{pe,nl}$ is an additional dependent variable that is solved as part of the global problem.

The structure of the local plasticity problem is not changed, and the numerical solution remains the same as discussed for the Numerical Integration Algorithm. The only difference is that the hardening force σ_{ys} now also depends on $\varepsilon_{pe,nl}$.

Resetting Plastic Variables

For some industrial applications, plastic strains are released due to manufacturing processes such as welding or stress relief annealing. To model these type of phenomena, it is possible to reset all or a subset of the internal plasticity variables to zero, or to prescribe them as user defined expressions or values.

Add a **Set Variables** subnode to Metal Plasticity, Porous Plasticity, Pressure-Dependent and Soil Plasticity, or Elastoplastic Soils to set up the plastic strain tensor, plastic deformation gradient, the equivalent plastic strain, and other internal variables.

METAL PLASTICITY

Permanent or plastic deformations in metals and other ductile materials is generally assumed to appear when the maximum shear stress or some other measure of distortion exceeds a critical value. This means that plastic yielding is independent of the hydrostatic pressure. Such models are often referred to as volume preserving plasticity models. The yield function is commonly described by the classical *von Mises* or *Tresca* criterion, but orthotropic plasticity may also be considered using the *Hill* criterion.

For a general discussion on plasticity and its implementation in COMSOL Multiphysics, see The Plasticity Problem section.

All plasticity models described in the following sections can be combined with different hardening models including:

- Isotropic Hardening

- Kinematic Hardening
- Mixed Hardening

von Mises Criterion

The *von Mises* criterion suggests that yielding of the material occurs when the second deviatoric stress invariant J_2 reaches a critical value k. Hence, von Mises plasticity is often also referred to as *J2 plasticity*. It can be written in terms of the elements of Cauchy's stress tensor (Ref. 54)

$$J_2 = \frac{1}{6}((\sigma_{11}-\sigma_{22})^2+(\sigma_{22}-\sigma_{33})^2+(\sigma_{33}-\sigma_{11})^2)+\sigma_{12}^2+\sigma_{23}^2+\sigma_{13}^2 = k^2$$

The von Mises criterion is implemented with the yield function

$$F_y = \sqrt{3J_2}-\sigma_{ys}$$

where σ_{ys} is the uniaxial *yield stress level* and $\sigma_{mises} = \sqrt{3J_2}$ is the equivalent von Mises stress. An associative flow rule is always used for the von Mises criterion, that is, $Q_p = F_y$

The equivalent von Mises stress σ_{mises} is available in the variable `solid.mises`, where `solid` is the name of the physics interface node.

Graphically, the von Mises yield surface is represented by an infinite cylinder in principal stress space with its axis coinciding with the hydrostatic axis. The elastic domain corresponds to the inside of the cylinder.

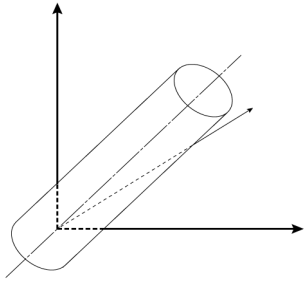


Figure 3-14: Graphical representation of the von Mises yield surface in principal stress space.

Tresca Criterion

The *Tresca* criterion suggests that yielding occurs when the maximum shear stress reaches a critical value. Given a spectral decomposition of Cauchy's stress tensor, the maximum shear stress is expressed as

$$\tau_{max} = \frac{1}{2}(\sigma_1 - \sigma_3)$$

where σ_1 is the maximum principal stress and σ_3 is the minimum principal stress. The Tresca criterion then reads

$$\tau_{max} = \tau_{ys}$$

where τ_{ys} is the shear yield stress. Alternatively, it can be expressed using the uniaxial yield stress σ_{ys}, as for the von Mises criterion, and then reads

$$2\tau_{max} = \sigma_1 - \sigma_3 = \sigma_{ys}$$

COMSOL Multiphysics uses the latter representation, and the Tresca yield function is implemented as

$$F_y = \sigma_{tresca} - \sigma_{ys} \tag{3-63}$$

where $\sigma_{tresca} = \sigma_1 - \sigma_3$ is the equivalent Tresca stress.

The Tresca equivalent stress, $\sigma_{tresca} = \sigma_1 - \sigma_3$, is implemented in the variable `solid.tresca`, where `solid` is the name of the physics interface node.

Graphically, the Tresca yield surface is represented by an infinite hexagonal prism in principal stress space with it axis coinciding with the hydrostatic axis, see Figure 3-15. The elastic domain corresponds to the inside of the hexagon. The shape of the hexagonal yield surface can be further visualized by projecting it onto the Octahedral

Plane as shown in Figure 3-16. The upper limit in this surface corresponds to the von Mises yield surface.

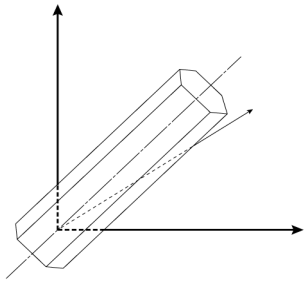


Figure 3-15: Graphical representation of the Tresca yield surface in principal stress space.

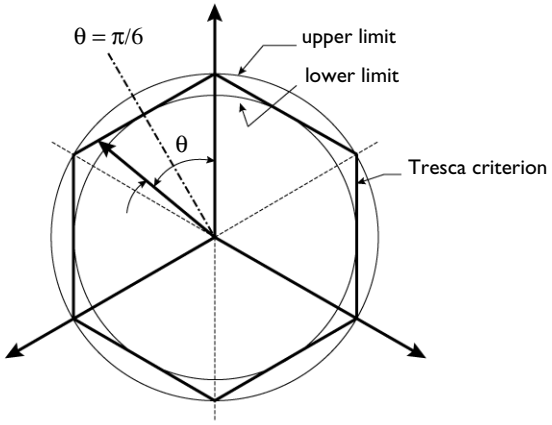


Figure 3-16: Graphical representation of the Tresca yield surface projected on the octahedral plane.

The Tresca criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow, the plastic potential equals the von Mises yield function, that is

$$Q_{\mathrm{p}} = \sqrt{3J_2}$$

This means that plastic flow always occurs in the radial direction toward the hydrostatic axis, and not perpendicular to the yield surface. The simplicity of this plastic potential improves the robustness and reduces the computational cost of the model.

For the case of associated plastic flow, COMSOL Multiphysics constructs a smooth plastic potential to handle the edges of the hexagonal yield function, by using the procedures outlined in Multisurface Plasticity. In general, a full representation of the hexagon requires six yield functions combined into one. However, since the principal stresses are sorted such that $\sigma_1 > \sigma_2 > \sigma_3$, it is sufficient to consider three functions only

$$Q_{\mathrm{p}}^1 = \sigma_1 - \sigma_3$$
$$Q_{\mathrm{p}}^2 = \sigma_2 - \sigma_3$$
$$Q_{\mathrm{p}}^3 = \sigma_1 - \sigma_2$$

Note that the exact representation of the yield function in Equation 3-63 is used for the associated plastic flow.

Hill Criterion

Hill (Ref. 96, Ref. 97) proposed a quadratic yield function given by the principal axes of orthotropy $\mathbf{a}_i$ in a local coordinate system. The yield function can be written in terms of the elements of Cauchy's stress tensor as

$$F_{\mathrm{y}} = F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L\sigma_{23}^2 + 2M\sigma_{13}^2 + 2N\sigma_{12}^2 - 1$$

By using a matrix notation, this reads

$$F_{\mathrm{y}} = \sigma^T P_{\mathrm{hill}} \sigma - 1$$

Here, σ is the vector representation of the stress tensor in Voigt order, see Orthotropic and Anisotropic Materials. The symmetric 6-by-6 matrix with Hill's coefficients reads

$$P_{\text{hill}} = \begin{bmatrix} G+H & -H & -G & 0 & 0 & 0 \\ -H & F+H & -F & 0 & 0 & 0 \\ -G & -F & F+G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2L & 0 & 0 \\ 0 & 0 & 0 & 0 & 2M & 0 \\ 0 & 0 & 0 & 0 & 0 & 2N \end{bmatrix}$$

where the parameters F, G, H, L, M, and N are related to the state of anisotropy.

Hill plasticity is an extension of *J2* (von Mises) plasticity, in the sense that it is volume preserving. Due to this property, six parameters are needed to define orthotropic plasticity, as opposed to orthotropic elasticity, where nine elastic coefficients are needed.

To write the Hill criterion in a similar way as done for the von Mises and Tresca criteria, an average initial yield stress σ_{ys0} can be defined from the parameters F, G, and H as the reciprocal average (Ref. 97)

$$\frac{1}{\sigma_{ys0}^2} = \frac{2}{3}(F+G+H)$$

or σ_{ys0} can be the uniaxial yield stress in one of the axes of orthotropy $\mathbf{a}_i$.

We can then define the dimensionless parameters f, g, h, l, m, and n as

$$p_{\text{hill}} = \sigma_{ys0}^2 P_{\text{hill}} = \begin{bmatrix} g+h & -h & -g & 0 & 0 & 0 \\ -h & f+h & -f & 0 & 0 & 0 \\ -g & -f & f+g & 0 & 0 & 0 \\ 0 & 0 & 0 & 2l & 0 & 0 \\ 0 & 0 & 0 & 0 & 2m & 0 \\ 0 & 0 & 0 & 0 & 0 & 2n \end{bmatrix}$$

Hill equivalent stress is then defined as

$$\sigma_{\text{hill}} = \sqrt{\sigma^T p_{\text{hill}} \sigma}$$

and the yield function is written as

$$F_y = \sigma_{\text{hill}} - \sigma_{ys}$$

with $\sigma_{ys} = \sigma_{ys0}$ in the absence of Isotropic Hardening.

Hill criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, Hill's plastic potential equals the von Mises yield function, that is

$$Q_p = \sigma_{mises} = \sqrt{3J_2}$$

which means that the plastic flow in stress space always occurs in the radial direction in the Octahedral Plane, and not normal to the yield surface given by Hill's criterion. The simplicity of von Mises plastic potential improves the robustness and reduces the computational cost of the model.

The equivalent plastic strain can also be associated or nonassociated. For nonassociated plasticity, the definition of the von Mises equivalent strain is used in the Hardening Rule

$$\dot{\varepsilon}_{pe} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{pl} : \dot{\varepsilon}_{pl}}$$

while for associated plasticity, Hill's equivalent plastic strain is computed from

$$\dot{\varepsilon}_{pe} = \sqrt{\frac{f\dot{\varepsilon}_{pl11}^2 + g\dot{\varepsilon}_{pl22}^2 + h\dot{\varepsilon}_{pl33}^2}{fg + fh + gh} + \frac{2}{l}\dot{\varepsilon}_{pl23}^2 + \frac{2}{m}\dot{\varepsilon}_{pl13}^2 + \frac{2}{n}\dot{\varepsilon}_{pl12}^2} = \lambda$$

Expressions for Hill's Coefficients The parameters in Hill criterion can be related to the initial yield stresses with respect to the axes of orthotropy $\mathbf{a}_i$. The first three parameters are related to the initial uniaxial yield stresses σ_{ys1}, σ_{ys2}, and σ_{ys3} as

$$2F = \frac{1}{\sigma_{ys2}^2} + \frac{1}{\sigma_{ys3}^2} - \frac{1}{\sigma_{ys1}^2}$$
$$2G = \frac{1}{\sigma_{ys3}^2} + \frac{1}{\sigma_{ys1}^2} - \frac{1}{\sigma_{ys2}^2}$$
$$2H = \frac{1}{\sigma_{ys1}^2} + \frac{1}{\sigma_{ys2}^2} - \frac{1}{\sigma_{ys3}^2}$$

and the remaining three parameters are related to the initial shear yield stresses σ_{ys12}, σ_{ys13}, and σ_{ys23} as

$$2L = \frac{1}{\sigma_{ys23}^2}$$
$$2M = \frac{1}{\sigma_{ys13}^2}$$
$$2N = \frac{1}{\sigma_{ys12}^2}$$

The reciprocal average yield stress then reads

$$\frac{1}{\sigma_{ys0}^2} = \frac{2}{3}(F+G+H) = \frac{1}{3}\left(\frac{1}{\sigma_{ys1}^2} + \frac{1}{\sigma_{ys2}^2} + \frac{1}{\sigma_{ys3}^2}\right)$$

When using the dimensionless parameters f, g, h, l, m, and n, the uniaxial yield stress in the principal axis of orthotropy $\mathbf{a_1}$ is used as reference yield stress.

For metal sheets, the principal axis of orthotropy is know as the *rolling direction* (RD), and the yield stress in this direction is denoted as σ_0.

When modeling metal sheets, the Lankford or anisotropic coefficients are r_0, r_{90}, and r_{45}, where the subindex denotes the angle with respect to the rolling direction. The dimensionless Hill's coefficients are then related to the anisotropic coefficients as

$$f = \frac{r_0}{r_{90}(1+r_0)}$$
$$g = \frac{1}{1+r_0}$$
$$h = \frac{r_0}{(1+r_0)}$$

In this case, the dimensionless parameters are related as $g + h = 1$.

The shear stresses σ_{ys13} and σ_{ys23} are difficult to measure, so it is typically assumed the same values as for isotropic plasticity

$$\sigma_{ys13} = \sigma_{ys23} = \frac{\sigma_0}{\sqrt{3}}$$

so the dimensionless coefficients read $l = m = 3/2$.

The parameter related to the in-plane shear stress, n, written as function of the anisotropic coefficients reads

$$n = \frac{(r_0 + r_{90})(0{,}5 + r_{45})}{r_{90}(1 + r_0)}$$

PRESSURE-DEPENDENT AND SOIL PLASTICITY

The yielding of many engineering materials is sensitive to the hydrostatic pressure. Such materials include biological tissue, polymers, foams, clays, and geotechnical materials. Assuming isotropic plastic deformation, the plasticity models for a large class of such materials can be constructed from the Invariants of the Stress Tensor. Here we use the following three invariants:

$$\sigma_m = \frac{I_1}{3}$$
$$\sigma_{mises} = \sqrt{3J_2}$$
$$\cos 3\theta = \frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{3/2}}$$

Here, σ_m is the *mean stress*, σ_{mises} is the von Mises *equivalent stress*, and θ is the *Lode angle*.

The **Pressure-Dependent Plasticity** and **Soil Plasticity** features implement a number of plasticity models that can be described by a yield function of either a parabolic form

$$F_y = \Gamma(\theta)\sigma_{mises} + M_a(\sigma_m) - \sigma_{ys} \tag{3-64}$$

or an elliptic form

$$F_y = \Gamma(\theta)^2\sigma_{mises}^2 + M_b(\sigma_m) - \sigma_{ys}^2 \tag{3-65}$$

Here, $\Gamma(\theta)$ is a function of the *Lode angle* θ that defines the shape of the yield function in the Octahedral Plane. The most common choice, $\Gamma(\theta) = 1$, results in a circular section. Similarly, $M_a(\sigma_m)$ and $M_b(\sigma_m)$ are functions of the *mean stress* σ_m that define the shape of the yield function in the Meridional Plane.

The yield stress σ_{ys} is here interpreted as the stress level when yielding occurs given $\Gamma(\theta) = 1$ and $\sigma_m = 0$. Various Isotropic Hardening models can be added by specifying the yield stress σ_{ys} as a function of the equivalent plastic strain ε_{pe}.

For a general discussion on plasticity models and their numerical implementation in COMSOL Multiphysics, see The Plasticity Problem section.

Drucker–Prager Criterion

The Drucker–Prager criterion is as an extension of the von Mises Criterion to include pressure-dependence. It is available in the **Porous Plasticity**, **Pressure-Dependent Plasticity**, and **Soil Plasticity** features and it can be used to model a wide variety of materials.

The resulting yield function is the simplest specialization of Equation 3-64, where the Octahedral Section is circular, $\Gamma(\theta) = 1$, and $M_a(\sigma_m) = \alpha_1\sigma_m$. The yield function reads

$$F_y = \sigma_{mises} + \alpha_1\sigma_m - \sigma_{ys}$$

The parameter α_1 governs how the criterion responds to changes in pressure. In this expression, the mean stress σ_m corresponds to the negative pressure, that is, it is positive in tension.

Graphically, the Drucker–Prager yield surface is represented by a cone in principal stress space with it axis aligned with the hydrostatic axis. The elastic domain is defined as the region enclosed by the cone.

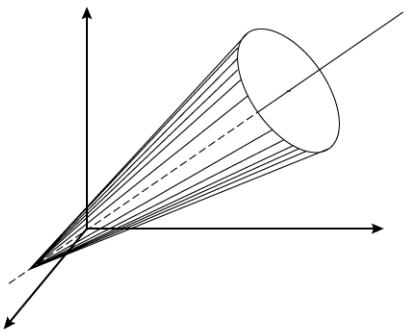


Figure 3-17: Graphical representation of the Drucker–Prager yield surface in stress space. The cone opens toward the compressive axis.

The Drucker–Prager criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the plastic potential is defined as

$$Q_p = \sigma_{mises} + a_{1Q}\sigma_m \tag{3-66}$$

The parameter a_{1Q} controls the influence of pressure on the plastic flow.

For **Soil Plasticity**, the parameters in Drucker–Prager criterion receive different names. The yield function is written as

$$F_y = \sigma_{mises} + \alpha\sigma_m - k$$

and the plastic potential for nonassociated flow as

$$Q_p = \sqrt{3J_2 + \sigma_{v,off}^2 \alpha_Q^2} + \alpha_Q \sigma_m$$

The parameters k, and α, are generally identified from common soil parameters: cohesion c; friction angle ϕ; and dilatation angle ψ, and by matching the Drucker–Prager criterion to the Mohr–Coulomb Criterion. This can be done using different assumptions:

- Matching at the compressive meridian ($\theta = \pi/3$). This means that the Drucker–Prager criterion circumscribes the Mohr–Coulomb criterion.

$$\alpha = \frac{6\sin\phi}{3 - \sin\phi},\ \alpha_Q = \frac{6\sin\psi}{3 - \sin\psi},\ \text{and}\ k = c\frac{6\cos\phi}{3 - \sin\phi}$$

- Matching at the tensile meridian ($\theta = 0$).

$$\alpha = \frac{6\sin\phi}{3 + \sin\phi},\ \alpha_Q = \frac{6\sin\psi}{3 + \sin\psi},\ \text{and}\ k = c\frac{6\cos\phi}{3 + \sin\phi}$$

- Inscribe the Mohr–Coulomb criterion. This is often used when plane strain conditions apply.

$$\alpha = \frac{3\sqrt{3}\tan\psi}{\sqrt{9 + 12\tan^2\phi}},\ \alpha_Q = \frac{3\sqrt{3}\tan\psi}{\sqrt{9 + 12\tan^2\psi}},\ \text{and}\ k = c\frac{3\sqrt{3}}{\sqrt{9 + 12\tan^2\phi}}$$

- Matching the uniaxial compressive and tensile strengths. This is useful when plane stress conditions apply.

$$\alpha = 9\sin\phi,\ \alpha_Q = 9\sin\psi, \text{ and } k = 2c\cos\phi$$

- Matching the biaxial compressive and tensile strengths. This is useful when plane stress conditions apply

$$\alpha = \frac{9}{2}\sin\phi,\ \alpha_Q = \frac{9}{2}\sin\psi, \text{ and } k = 2c\cos\phi$$

See also the description of the Drucker–Prager model in the Solid Mechanics interface documentation.

Apex Smoothing

The plastic potential in Equation 3-66 leads to a singularity in the flow rule where the yield surface intersects the hydrostatic axis, that is, the apex of the cone. Such formulation is not suitable for a robust numerical implementation. To handle this, COMSOL Multiphysics modifies Equation 3-66 such that a smooth plastic potential is obtained with a rounded apex. The smooth plastic potential is written as

$$Q_p = \sqrt{3J_2 + \sigma_{v,off}^2 a_{1Q}^2} + a_{1Q}\sigma_m \tag{3-67}$$

Here, the smoothing parameter $\sigma_{v,off}$ controls the distance where the potential in Equation 3-66 and the smooth potential in Equation 3-67 intersect the hydrostatic axis as shown in Figure 3-18.

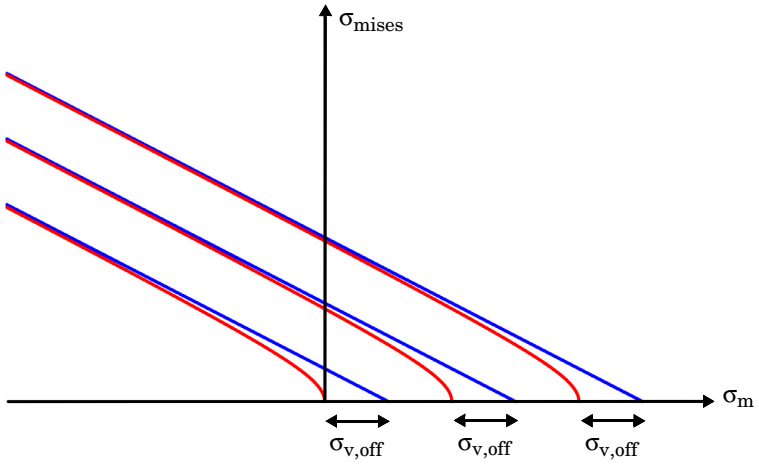


Figure 3-18: Isolines of the plastic potential in on the meridional plane. Blue lines show the original Drucker–Prager potential and red lines show the smooth potential.

For **Soil Plasticity**, the smooth plastic potential for nonassociated flow reads

$$Q_p = \sqrt{3J_2 + \sigma_{v,off}^2 \alpha_Q^2} + \alpha_Q \sigma_m$$

Elliptic Criterion

The elliptic criterion in **Pressure-Dependent Plasticity** implements Equation 3-65 with the meridional shape function parameterized with a linear and a quadratic dependence on the mean stress σ_m as

$$M_b(\sigma_m) = b_1 \sigma_m + b_2^2 (\sigma_m - b_3)^2$$

The shape of the yield function in the Octahedral Plane is generally circular, but it can be made a function of the Lode angle θ by using any of the functions described in Octahedral Section.

Setting $b_1 = 0$ recovers an ellipse like in for example the Cam–clay family of models. Parameter $b_3 > 0$ then represents the shift of the ellipse along the hydrostatic axis, and parameter b_2 gives the aspect ratio of the ellipse in the Meridional Plane.

Setting $b_2 = 0$ and $b_1 > 0$ results into a paraboloid that opens into the compressive direction and intersects the hydrostatic axis at $\sigma_m = \sigma_{ys}^2/b_1$. Setting $b_1 = 0$ and $b_2 = 0$ recovers the von Mises Criterion.

The elliptic criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the plastic potential is defined as

$$Q_p = \sigma_{mises}^2 + b_{1Q}\sigma_m + b_{2Q}^2(\sigma_m - b_{3Q})^2$$

with a set of three parameters to control the shape of the plastic potential in the Meridional Plane. For nonassociated flow, the plastic potential is always assumed to be independent of the Lode angle θ, and its projection on the Octahedral Plane is thus circular.

Parabolic Criterion

The parabolic criterion in **Pressure-Dependent Plasticity** implements Equation 3-64 with the meridional shape function parameterized with a linear, a quadratic, and an exponential dependency on the mean stress σ_m as

$$M_a(\sigma_m) = a_1\sigma_m + a_2\sigma_m^2 + a_3\exp(a_4\sigma_m)$$

The shape of the yield function in the Octahedral Plane is generally circular, but it can be made as a function of the Lode angle θ by using any of the functions described in Octahedral Section.

Setting $a_1 > 0$, $a_2 = a_3 = 0$ recovers the Drucker–Prager Criterion. By setting $a_2 > 0$, $a_1 = a_3 = 0$, a closed parabola is obtained similar to, for example, the GAZT criterion for foams.

The parabolic criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the plastic potential is defined as

$$Q_p = \sigma_{mises} + a_{1Q}\sigma_m + a_{2Q}\sigma_m^2 + a_{3Q}\exp(a_{4Q}\sigma_m) \tag{3-68}$$

with a set of four parameters to control the shape of the plastic potential in the Meridional Plane. For nonassociated flow, the plastic potential is always assumed to be independent of the Lode angle θ, and its projection onto the Octahedral Plane is thus circular.

The plastic potential in Equation 3-68 leads to a singularity in the flow rule where the yield surface intersects the hydrostatic axis, that is, the apex of the cone. This formulation is not suitable for a robust numerical implementation. To handle this,

COMSOL Multiphysics implements Equation 3-68 with a smooth plastic potential with a rounded apex. The smooth plastic potential is written as

$$Q_{\mathrm{p}} = \sqrt{3J_2 + \sigma_{\mathrm{v,off}}^2 a_{1\mathrm{Q}}^2} + a_{1\mathrm{Q}}\sigma_{\mathrm{m}} + a_{2\mathrm{Q}}\sigma_{\mathrm{m}}^2 + a_{3\mathrm{Q}}\exp(a_{4\mathrm{Q}}\sigma_{\mathrm{m}}) \tag{3-69}$$

Here, the smoothing parameter $\sigma_{\mathrm{v,off}}$ controls the distance between where the potential in Equation 3-68 and the potential in Equation 3-69 intersect the hydrostatic axis. If $\alpha_{1\mathrm{Q}} = 0$, a small fall-back value is used in the square root term of Equation 3-69. The smooth plastic potential is used for the *associated plastic flow* also. See Apex Smoothing for details.

Octahedral Section

The profile of the yield function in the Octahedral Plane is useful for denoting changes in the yield limit under different loading conditions, for example, compression or tension. COMSOL Multiphysics provides three alternative definitions of the function $\Gamma(\theta)$ that controls the shape of the yield function on the octahedral plane:

- Circular, $\Gamma(\theta) = 1$
- Gudehus
- Lagioia–Panteghini

See Octahedral Section for details.

Foam Criterion

The foam plasticity model in **Pressure-dependent plasticity**, is an elliptic model that is a generalization of the model first proposed by Deshpande and Fleck (Ref. 119) to model structural foams. It is popular not only for modeling the failure of metallic foams, but also for battery cells, biological tissue, and other materials with a cellular structure.

The COMSOL Multiphysics implementation of the model can be seen as a specialization of Equation 3-65 with the yield function written as

$$F_y = \sigma_{\mathrm{mises}}^2 + \alpha^2\left(\sigma_{\mathrm{m}} - \frac{p_{\mathrm{t}} - p_{\mathrm{c}}}{2}\right)^2 - \left(\alpha\frac{p_{\mathrm{t}} - p_{\mathrm{c}}}{2}\right)^2 \tag{3-70}$$

The shape of the ellipse is controlled by three parameters, where p_{t} and p_{c} define the limits in hydrostatic tension and compression, respectively. The parameter α controls the aspect ratio of the ellipse, and it is inferred from the initial uniaxial compressive strength ${\sigma_{\mathrm{uc}}}^0$ such that

$$\alpha = \frac{3\frac{p_t}{p_c^0}}{\sqrt{\left(3\frac{\sigma_{uc}^0}{p_c^0} + \frac{p_t}{p_c^0}\right)\left(3 - \frac{p_t}{p_c^0}\right)}}$$

Here, p_c^0 is the initial yield stress in hydrostatic compression when hardening is considered, which means that the aspect ratio of the ellipse is constant after plastic deformation.

The foam plasticity model always uses a nonassociated flow rule with the plastic potential defined as

$$Q_p = \sigma_{mises}^2 + \beta^2 \sigma_m^2 \tag{3-71}$$

Compared to the yield function in Equation 3-70, the centroid of the ellipse defined by the plastic potential in Equation 3-71 is located at the origin of the Meridional Plane. The shape of the plastic potential is defined by a single parameter, β, that controls the aspect ratio of the ellipse. The parameter is defined by the *plastic Poisson's ratio* ν_p as

$$\beta = \frac{3}{\sqrt{2}}\sqrt{\frac{1-2\nu_p}{1+\nu_p}}$$

The plastic Poisson's ratio is identified as the ratio of longitudinal to transverse plastic strain, and has a lower limit of -1 and an upper limit of 0.5. A common assumption for metallic foams is $\nu_p = 0$, which results in a parameter $\beta = 9/2$.

The foam plasticity model is intended to model the compaction of materials and thus adds a volumetric hardening model. The hardening rule is implemented for the evolution of the internal plastic variable ε_{pv}, which is the *equivalent volumetric plastic strain*. The hardening rule is set up so that ε_{pv} is monotonically increasing and it only accounts for the compaction of the material

$$\dot{\varepsilon}_{pv} = -\langle -\mathrm{tr}\dot{\varepsilon}_{pl}\rangle = -\lambda\langle -\mathrm{tr}G_{pl}\rangle$$

Note that ε_{pv} is positive in compression.

The volumetric hardening model then defines the compressive hydrostatic yield stress as

$$p_c = p_c^0 + p_c^h(\varepsilon_{pv})$$

where p_c^h is a hardening function. A common hardening model for foams is that the compressive yield stress varies with the volumetric strain by a power function, that is, $p_c^h(\varepsilon_{pv}) = C\,\varepsilon_{pv}^n$.

Test data is sometimes more commonly available in terms uniaxial compressive stress and strain. COMSOL Multiphysics then also provides the option to specify hardening data in terms of the uniaxial compressive strength σ_{uc} and the axial equivalent plastic strain ε_{pa} as

$$\sigma_{uc} = \sigma_{uc}^0 + \sigma_{uc}^h(\varepsilon_{pa})$$

From the uniaxial compressive strength, the compressive hydrostatic yield stress is computed as

$$p_c = \frac{\frac{\sigma_{uc}}{\alpha^2} - \frac{\sigma_{uc}}{9} + \frac{p_t}{3}}{p_t + \frac{\sigma_{uc}}{3}}$$

The axial equivalent plastic strain ε_{pa} is difficult to identity for general loading conditions. It is therefore estimated from the equivalent volumetric plastic strains ε_{pv} and the plastic Poisson's ratio ν_p as

$$\varepsilon_{pa} = \frac{\varepsilon_{pv}}{1 - 2\nu_p}$$

Note that when volumetric hardening is included, it is only the yield stress in hydrostatic compression p_c that evolves with the plastic deformation. All other parameters that define the yield function and plastic potential remain fixed.

Mohr–Coulomb Criterion

The Mohr–Coulomb criterion is a classical and popular criterion in soil mechanics. It was developed by Coulomb before the Tresca and von Mises criteria for Metal Plasticity, and it was the first criterion to include the hydrostatic pressure. The criterion states that failure occurs when the shear stress and the normal stress acting on any element in the material satisfy the equation

$$|\tau| + \sigma\tan\phi - c = 0$$

were, τ is the shear stress, c the cohesion, and ϕ denotes the friction angle.

With the help of Mohr's circle, the criterion can be written as

$$\frac{1}{2}(\sigma_1 - \sigma_3) + \frac{1}{2}(\sigma_1 + \sigma_3)\sin\phi - c\cos\phi = 0$$

The Mohr–Coulomb criterion defines an irregular hexagonal pyramid in principal stress space.

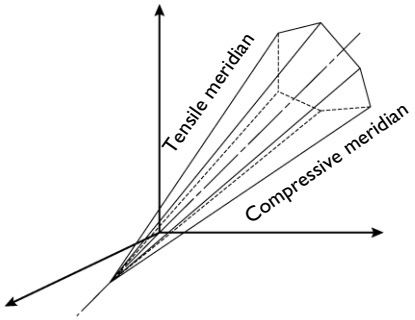


Figure 3-19: The Mohr–Coulomb criterion. The cone opens up toward the compressive axis.

The Mohr–Coulomb criterion can be written in terms of invariants as a specialization of the general yield function presented in Equation 3-64. Since the yield function has a hexagonal octahedral section, it is written as

$$F_y = \Gamma(\theta)\sigma_{\text{mises}} + \alpha\sigma_{\text{m}} - k$$

Here parameters α and k are defined by the friction angle ϕ and the cohesion c as

$$\alpha = \frac{6\sin\phi}{3 - \sin\phi} \text{ and } k = c\frac{6\cos\phi}{3 - \sin\phi}$$

To define an octahedral section that equals the hexagonal shape of the Mohr–Coulomb criterion, the function in Equation 3-69 is used.

The Mohr–Coulomb criterion can be used with either an associated or nonassociated flow rule. For the associated flow rule, $\alpha_Q = \alpha$ such that the pressure dependence of plastic flow is governed by the friction angle ϕ. When a nonassociated flow rule is used, α_Q is defined by the dilatation angle ψ as

$$\alpha_Q = \frac{6\sin\psi}{3-\sin\psi}$$

It is also possible to use a simplified flow rule with the plastic potential defined by Equation 3-67, that is, independent of the Lode angle θ.

See also the description of the Mohr–Coulomb model in the Solid Mechanics interface documentation.

Vertex Smoothing

To define the Mohr–Coulomb yield function, the three parameters in Equation 3-69 are defined as

$$c_1 = \sec\left[(\bar{\gamma}+1)\frac{\pi}{6}\right],\ c_2 = 1, \text{ and } c_3 = 1-\bar{\gamma}$$

with the intermediate variable

$$\bar{\gamma} = \frac{6}{\pi}\operatorname{atan}\frac{\sin\phi}{\sqrt{3}}$$

The choice of these parameters results in the hexagonal pyramid depicted in Figure 3-19 with the sharp corners. If used as plastic potential for the flow rule, this leads to singularities and a formulation not suitable for a robust numerical implementation. COMSOL Multiphysics defines the smooth plastic potential as

$$Q_p = \sqrt{\Gamma(\theta)^2\sigma_{\text{mises}}^2 + \sigma_{\text{v,off}}^2\alpha_Q^2} + \alpha_Q\sigma_m \qquad (3\text{-}72)$$

where the apex is rounded using the same method as described in Apex Smoothing.

The sharp vertices on the octahedral section are rounded by setting $c_2 = \beta$ with a smoothing parameter $\beta < 1$ in the nonassociated plastic potential. This results in a smooth and continuous octahedral section that inscribes the Mohr–Coulomb criterion.

The default value is $\beta = 0.99$. Setting a lower value for β increases the radius of the rounded corners.

Matsuoka–Nakai Criterion

Matsuoka and Nakai (Ref. 58) discovered that the sliding of soil particles occurs in the plane in which the ratio of shear stress to normal stress has its maximum value, which

they called the *mobilized plane*. The yield surface is based on the first, second, and third Invariants of the Stress Tensor as

$$(9 + 9\mu^2)I_3 - I_1 I_2 = 0$$

where the parameter $\mu = (\tau/\sigma_n)_{STP}$ equals the maximum ratio between shear stress and normal stress in the *spatially mobilized plane* (STP-plane). The invariants are applied over the effective stress tensor, this is, the stress tensor minus the fluid pore pressure.

The Matsuoka–Nakai criterion circumscribes the Mohr–Coulomb Criterion when the parameter μ equals

$$\mu = \frac{2\sqrt{2}}{3}\tan\phi$$

where ϕ denotes the friction angle.

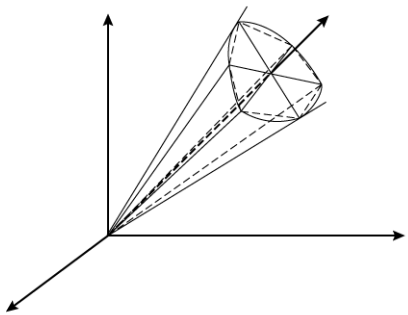


Figure 3-20: The Matsuoka–Nakai and Mohr–Coulomb criteria in the principal stress space.

As shown in Ref. 100, the above formulation of the Matsuoka–Nakai criterion is not suitable for numerical implementation since it results in false elastic domains in the tensile octant's of the principal stress space.

COMSOL Multiphysics implements the Matsuoka–Nakai criterion as a specialization of the general yield function presented in Equation 3-64. Since the yield function has a noncircular octahedral section, it is written as

$$F_y = \Gamma(\theta)\sigma_{mises} + \alpha\sigma_m - k$$

Here, the parameters α and k can be defined by the friction angle ϕ and the cohesion c as

$$\alpha = \frac{6\sin\phi}{3-\sin\phi} \text{ and } k = c\frac{6\cos\phi}{3-\sin\phi}$$

To define an octahedral section that equals the Matsuoka–Nakai criterion, the function proposed by Panteghini and Lagioia (Ref. 100) in Equation 3-69 is used. Here, the three parameters of Equation 3-69 are defined as

$$c_1 = \frac{2}{3}\sqrt{A_1}\alpha,\ c_2 = \frac{A_2}{A_1^{3/2}}, \text{ and } c_3 = 0$$

with the intermediate variables

$$k_{MN} = (9 + 9\mu^2),\ A_1 = \frac{k_{MN}-3}{k_{MN}-9}, \text{ and } A_2 = \frac{k_{MN}}{k_{MN}-9}$$

As for all parabolic yield functions, the Matsuoka–Nakai yield function has a singularity where it intersects the hydrostatic axis. Therefore, the same smooth plastic potential as for the Mohr–Coulomb criterion (Equation 3-72) is used.

The Matsuoka–Nakai criterion can be used with either an associated or nonassociated flow rule. For the associated flow rule, $\alpha_Q = \alpha$, such that the pressure dependence of plastic flow is governed by the friction angle ϕ. When a nonassociated flow rule is used, α_Q is defined by the dilatation angle ψ as

$$\alpha_Q = \frac{6\sin\psi}{3-\sin\psi}$$

It is also possible to use a simplified flow rule with the plastic potential defined by Equation 3-67, that is, independent of the Lode angle θ.

See also the description of the Matsuoka–Nakai material model in the Solid Mechanics interface documentation.

Lade–Duncan Criterion

The Lade–Duncan criterion was originally developed to model a large volume of laboratory sample test data of cohesionless soils. The criterion is defined as

$$k_{\mathrm{LD}} I_3 - I_1^3 = 0$$

where I_1 and I_3 are the first and third Invariants of the Stress Tensor, respectively, and k_{LD} is a parameter related to the direction of the plastic strain increment in the triaxial plane. It can be derived from the friction angle ϕ as

$$k_{\mathrm{LD}} = \frac{(3 - \sin\phi)^3}{(1 + \sin\phi)(1 - \sin\phi)^2}$$

The Lade–Duncan criterion does not match the Mohr–Coulomb Criterion at the tensile meridian ($\theta = 0$).

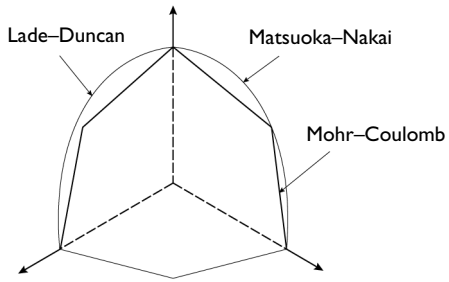


Figure 3-21: Comparing the Mohr–Coulomb, Matsuoka–Nakai, and Lade–Duncan criteria on the octahedral plane.

As shown in Ref. 100, the above formulation of the Lade–Duncan criterion is not suitable for numerical implementation since it results in false elastic domains in the tensile octants of the principal stress space.

COMSOL Multiphysics instead implements Lade–Duncan criterion as a specialization of the general yield function presented in Equation 3-64. Since the yield function has a noncircular octahedral section, it is written as

$$F_y = \Gamma(\theta)\sigma_{\mathrm{mises}} + \alpha\sigma_{\mathrm{m}} - k$$

Here, parameters α and k can be defined by the friction angle ϕ and the cohesion c as

$$\alpha = \frac{6\sin\phi}{3 - \sin\phi} \quad \text{and} \quad k = c\frac{6\cos\phi}{3 - \sin\phi}$$

To match an octahedral section corresponding to Lade–Duncan criterion, the function proposed by Panteghini and Lagioia (Ref. 100) is used. Here, the three parameters of Equation 3-69 are defined as

$$c_1 = \frac{2}{3}\sqrt{A_1}\alpha,\quad c_2 = \frac{1}{\sqrt{A_1}},\text{ and } c_3 = 0$$

with the intermediate variable

$$A_1 = \frac{k_{\mathrm{LD}}}{k_{\mathrm{LD}} - 27}$$

As for all parabolic yield functions, the Lade–Duncan yield function has a singularity where it intersects the hydrostatic axis. Therefore, the same smooth plastic potential as for the Mohr–Coulomb criterion (Equation 3-72) is used.

The Lade–Duncan criterion can be used with either an associated or nonassociated flow rule. For the associated flow rule, $\alpha_Q = \alpha$, such that the pressure dependence of plastic flow is governed by the friction angle ϕ. When a nonassociated flow rule is used, α_Q is defined by the dilatation angle ψ as

$$\alpha_Q = \frac{6\sin\psi}{3 - \sin\psi}$$

It is also possible to use a simplified flow rule with the plastic potential defined by Equation 3-67, that is, independent of the Lode angle θ.

See also the description of the Lade–Duncan model in the Solid Mechanics interface documentation.

Compression Cap

For many type of plasticity models, a cap can be added to limit the elastic domain in compression. This is of particular interest for models that are linear in the Meridional Plane such as the Drucker–Prager Criterion, and other soil plasticity models where the yield function opens in the compressive direction of the hydrostatic axis. Normally, these models are not accurate above a given pressure limit, since real-life materials cannot bear infinite loads and still behave elastically. To circumvent this, three types of cap models can be added in COMSOL Multiphysics

- Planar

- Elliptic
- Cam-clay

The *elliptic cap* formulation is the most comprehensive, often providing more accurate predictions of observed behavior. However, the simplicity of the *planar cap* and the *Cam-clay cap* make them practical alternatives. These formulations employ an associated flow rule and allow for hardening independently of the chosen yield function.

The combination of a general yield function with a cap formulation leads to an nonsmooth yield surface, which is unsuitable for numerical implementation. To overcome this, COMSOL Multiphysics uses the procedures outlined in Multisurface Plasticity to construct a smooth and C^2 differentiable yield surface.

Planar Cap The yield function of the planar cap only depends on the mean stress σ_m

$$F_{cap} = -\sigma_m - p_c$$

where p_c is the *pressure limit*. The yield function imposes a pressure limit that restricts any values exceeding p_c.

Isotropic hardening can be added, such that the pressure limit is defined as

$$p_c = p_{c0} + p_{ch}(\varepsilon_{pc})$$

where p_{c0} is the *initial pressure limit*, and p_{ch} is the *hardening function* that depends on the hardening variable ε_{pc}. A *perfectly plastic* model assumes $p_{ch} = 0$.

Elliptic Cap The yield function of the elliptic cap is a function of stress invariants σ_m, σ_{mises}, and θ; it only depends on the Lode angle if the parent yield function does. For the most general case, the yield function reads

$$F_{cap} = \sqrt{\Gamma(\theta)^2 3J_2 + \frac{1}{R^2}\langle -\sigma_m - p_{cc}\rangle^2} + F_y(-p_{cc}, 0, 0)$$

Where the operator $< \cdot >$ represents the Macaulay brackets. The term $F_y(-p_{cc},0,0)$ implies that the parent yield function (or failure criterion) is evaluated at $\sigma_m = -p_{cc}$, at zero von Mises stress, $\sigma_{mises} = 0$, and at zero Lode angle, $\theta = 0$.

The location and the shape of the elliptic cap is controlled by its centroid in the hydrostatic axis, p_{cc}, and the aspect ratio R. The latter is defined by

$$R = \frac{p_{cc0} - p_{c0}}{F_y(-p_{cc0}, 0, 0)}$$

where p_{c0} is the initial pressure limit and p_{cc0} is the initial location of the centroid on the hydrostatic axis. This definition implies that the aspect ratio is constant also when hardening is included.

Figure 3-22 visualizes the combined yield surface of the Drucker–Prager Criterion and the elliptic cap on the meridional plane.

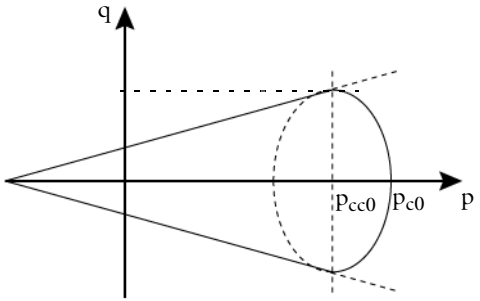


Figure 3-22: Elliptic cap model in the meridional plane.

A general hardening model can be added to the elliptic cap so that it evolves with plastic deformation. For the general case, the hardening model controls the centroid of the ellipse by

$$p_{cc} = p_{cc0} + p_{ch}(\varepsilon_{pc})$$

Here p_{ch} is the hardening function that depends on the hardening variable ε_{pc}. However, when the Drucker–Prager Criterion is used, the hardening model directly specifies the evolution of the pressure limit p_c as

$$p_c = p_{c0} + p_{ch}(\varepsilon_{pc})$$

where p_{c0} is the initial pressure limit and p_{ch} is the hardening function. The current value of the centroid variable, p_{cc}, is then computed from p_c using geometrical considerations.

Cam-Clay Cap The Cam-clay cap is similar to the elliptic cap, when the initial centroid of the ellipse, p_{cc0}, corresponds to half the initial pressure limit, p_{c0}.

$$p_{cc0} = \frac{p_{c0}}{2}$$

See also The Modified Cam-Clay Soil Model.

Cap Hardening The variable ε_{pc} is an internal plastic variable added when hardening is included. The additional hardening rule for this variable reads

$$\dot{\varepsilon}_{pc} = \begin{cases} \lambda h_c & \text{if } \mathrm{tr}\dot{\varepsilon}_{pl} < 0 \\ 0 & \text{else} \end{cases} \tag{3-73}$$

where h_c is the hardening modulus.

The hardening modulus is often defined as the rate of the volumetric plastic strain, that is the trace of the Plastic Flow Rule, $h_c = \mathrm{tr}\, G_{pl}$, but it can also be defined as a user defined expression.

The condition in Equation 3-73 means that the hardening variable ε_{pc} decreases monotonically with volumetric plastic deformation. However, it is possible to allow the cap to soften. The cap is then limited to move along the compressive part of the hydrostatic axis, and the hardening rule is modified to read

$$\dot{\varepsilon}_{pc} = \begin{cases} \lambda h_c & \text{if } \mathrm{tr}\dot{\varepsilon}_{pl} < 0 \text{ or } p_{cc} > 0 \\ 0 & \text{else} \end{cases}$$

When applying hardening to the planar cap, the variable for the centroid of the ellipse, p_{cc}, is replaced by the variable for the pressure limit, p_c.

Cap Hardening Functions The hardening functions p_{ch} and p_{cch} can be defined by user defined expressions. COMSOL Multiphysics also includes built-in hardening functions intended to model a variation of the material density during compaction.

For linear hardening, the hardening function reads

$$p_{ch}(\varepsilon_{pc}) = -K_c\varepsilon_{pc}$$

The exponential hardening function for the planar cap reads

$$p_{\mathrm{ch}}(\varepsilon_{\mathrm{pc}}) = -K_{\mathrm{c}}\ln\left(1+\frac{\varepsilon_{\mathrm{pc}}}{\varepsilon_{\mathrm{pvol}}^{\max}}\right)$$

and

$$p_{\mathrm{cch}}(\varepsilon_{\mathrm{pc}}) = -K_{\mathrm{c}}\ln\left(1+\frac{\varepsilon_{\mathrm{pc}}}{\varepsilon_{\mathrm{pvol}}^{\max}}\right)$$

for the elliptic and Cam-clay cap. For these formulations, K_{c} is the hardening modulus and $\varepsilon_{\mathrm{pvol}}^{\max}$ is the maximum volumetric plastic strain.

For a user defined hardening function, enter an expression for $p_{\mathrm{ch}}(\varepsilon_{\mathrm{pc}})$.

Tension Cutoff

In many plasticity models, particularly for soils, the predicted uniaxial tensile strength tends to overestimate experimental results. To improve the accuracy of these predictions, a tension cutoff can be introduced. Two types of cutoff models are typically used to address this issue

- Mean stress
- Rankine

In general, the Rankine cutoff offers a more accurate formulation, whereas the mean stress cutoff provides a more efficient and robust alternative. Both cutoff models utilize an associative flow rule. However, combining a general parabolic yield function with either cutoff formulation leads to an nonsmooth yield surface, which is unsuitable for numerical implementation. To overcome this drawback, COMSOL Multiphysics uses the procedures outlined in Multisurface Plasticity to construct a smooth and C^2 differentiable yield function.

Mean Stress Cutoff The yield function of the mean stress cutoff is only dependent on the mean stress σ_{m} and reads

$$F_{\mathrm{cutoff}} = \sigma_{\mathrm{m}} - \sigma_{\mathrm{m,t}}$$

where $\sigma_{\mathrm{m,t}}$ is the mean stress limit in tension. The yield function implies that a mean stress exceeding $\sigma_{\mathrm{m,t}}$ is not allowed.

Rankine Cutoff The Rankine criterion states that the maximum principal stress cannot exceed the tensile strength σ_{ts}. Since COMSOL Multiphysics sorts the principal stresses ($\sigma_1 > \sigma_2 > \sigma_3$), the criterion reads

$$\sigma_1 > \sigma_{ts}$$

However, the resulting yield function is neither smooth nor continuously differentiable. To construct a smooth Rankine yield function, COMSOL Multiphysics considers all three planes in stress space, that is

$$F^1_{cutoff} = \sigma_1 - \sigma_{ts}$$
$$F^2_{cutoff} = \sigma_2 - \sigma_{ts}$$
$$F^3_{cutoff} = \sigma_3 - \sigma_{ts}$$

and the applies the procedures outlined in Multisurface Plasticity to construct a smooth and C^2 differentiable yield function F_{cutoff}.

ISOTROPIC HARDENING

A plasticity model with isotropic hardening is characterized by a yield surface that expands (or shrinks) uniformly during plastic deformation. For example, the von Mises Criterion furbished with an isotropic hardening law implies that the radius of the circular cross section in the Octahedral Plane changes during plastic deformation, but its axis remain on the hydrostatic axis.

An isotropic hardening model typically considers a single scalar hardening variable. COMSOL Multiphysics uses the *equivalent plastic strain* ε_{pe} as described by the hardening rule in Equation 3-60. By default, most plasticity features assume a strain hardening approach to define ε_{pe}, but other definitions may also be considered.

Isotropic hardening is added to the general yield function in Equation 3-54 by letting the *yield stress* σ_{ys} be a function of the equivalent plastic strain ε_{pe}. That is, the yield function reads

$$F_y(\sigma, \sigma_{ys}(\varepsilon_{pe})) = 0$$

or in most cases

$$F_y = \sigma_e(\sigma) - \sigma_{ys}(\varepsilon_{pe})$$

where σ_e is the scalar equivalent stress. COMSOL Multiphysics provides several types of isotropic hardening models for elastoplastic materials:

- Perfect Plasticity
- Linear

- Ludwik
- Johnson–Cook
- Swift
- Voce
- Hockett–Sherby
- Hardening Function

The available built-in models vary for different features.

The internal variable for the equivalent plastic strain is named `solid.epe`. The equivalent plastic strain evaluated at Gauss points is named `solid.epeGp`.

Perfect Plasticity

For perfect (or ideal) elastoplastic materials, the yield surface is fixed in the space of principal stresses, and therefore, plastic deformations occur only when the stress path moves along the yield surface. That is, the plasticity model does not include isotropic hardening.

Here the yield stress is given by the *initial yield stress* σ_{ys0} such that it is constant and independent on the equivalent plastic strain variable ε_{pe}

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0}$$

Linear

The yield stress $\sigma_{ys}(\varepsilon_{pe})$ is for linear isotropic hardening a function of the *equivalent plastic strain* ε_{pe} and the *initial yield stress* σ_{ys0}

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + E_{iso}\varepsilon_{pe}$$

Here, the *isotropic hardening modulus* E_{iso} is derived from

$$\frac{1}{E_{iso}} = \frac{1}{E_{Tiso}} - \frac{1}{E}$$

For linear isotropic hardening, the *isotropic tangent modulus* E_{Tiso} is defined as (stress increment / total strain increment). The Young's modulus E is taken from the parent material model. If the elasticity of the parent material is anisotropic or orthotropic, E represents an equivalent Young's modulus.

Ludwik

In the Ludwik model for nonlinear isotropic hardening, the yield stress $\sigma_{ys}(\varepsilon_{pe})$ is defined by a nonlinear function of the equivalent plastic strain. The Ludwik equation (also called the Ludwik–Hollomon equation) for isotropic hardening is given by the power law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + k\varepsilon_{pe}^{n}$$

Here, k is the strength coefficient and n is the hardening exponent. Setting $n = 1$ would result in linear isotropic hardening.

Johnson–Cook

Similar to the Ludwik hardening model, the yield stress $\sigma_{ys}(\varepsilon_{pe})$ is in the Johnson–Cook model defined as a nonlinear function of the equivalent plastic strain ε_{pe} given by a power law. The difference is that effects of the strain rate are included in the model. The yield stress and hardening function is given by

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})\left(1 + C\log\left(\frac{\dot{\varepsilon}_{pe}}{\dot{\varepsilon}_{0}}\right)\right)(1 - f(T_{h})) \tag{3-74}$$

The Johnson–Cook model also adds the possibility to include thermal softening by including a function which depends on the *normalized homologous temperature* T_h. It is a function of the current temperature T, the melting temperature of the metal T_m, and a reference temperature T_{ref}

$$T_{h} = \frac{T - T_{ref}}{T_{m} - T_{ref}}$$

As a default, this dependency follows a power law, $f(T_h) = T_h^m$. The softening function $f(T_h)$ should have the properties $f(T_h < 0) = 0$ and $f(T_h > 1) = 1$.

!	The natural logarithm is used in Equation 3-74. If material data have been determined using a base 10 logarithm, the value of C must be adjusted accordingly.

The strain rate strength coefficient C and the reference strain rate $\dot{\varepsilon}_0$ in the Johnson–Cook model are obtained from experimental data. The flow curve properties are normally investigated by conducting uniaxial tension tests in a temperature range

around half of the melting temperature T_m, and subjected to different strain rates like 0.1/s, 1/s, and 10/s.

The original motivation of the Johnson–Cook is to model material failure at high strain rates, and it can be shown that the strain rate dependent term in Equation 3-74 is only valid when $\dot{\varepsilon}_{pe} > \dot{\varepsilon}_0$. The implementation of the Johnson–Cook model in COMSOL Multiphysics thus reads

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})\left[1 + C\log\left(1 + \left\langle \frac{\dot{\varepsilon}_{pe}}{\dot{\varepsilon}_0} - 1 \right\rangle\right)\right](1 - f(T_h))$$

where the operator $\langle \cdot \rangle$ represents the Macaulay brackets, such that the logarithmic function never evaluates to a negative number. In this formulation, the reference strain rate $\dot{\varepsilon}_0$ must be considered in the quasistatic limit. Below this value, the hardening function will be strain rate independent, but temperature dependent, similar to Ludwik law:

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})(1 - f(T_h)) \qquad \text{for} \qquad \dot{\varepsilon}_{pe} < \dot{\varepsilon}_0$$

an alternative implementation is provided for the case when hardening must account for both high and low strain rates, possibly in the same material point as the loading changes. This yield stress for this *modified* Johnson–Cook model is given by

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})\left[1 + C\log\left(1 + \frac{\dot{\varepsilon}_{pe}}{\dot{\varepsilon}_0}\right)\right](1 - f(T_h))$$

which is valid for $\dot{\varepsilon}_{pe} > 0$. At high strain rates where $\dot{\varepsilon}_{pe} \gg \dot{\varepsilon}_0$ it approaches the original model in Equation 3-74, while for $\dot{\varepsilon}_{pe} < \dot{\varepsilon}_0$ the strain rate dependence becomes similar to Peric Model.

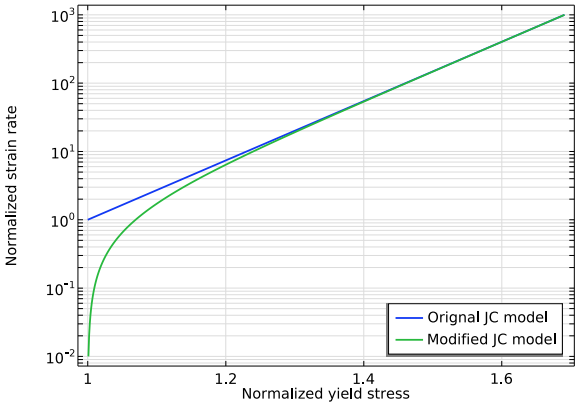


Figure 3-23: Comparison of the strain rate effect of the original and modified Johnson–Cook models.

Swift

For nonsaturating materials, the Swift power law equation relates the initial yield stress σ_{ys0} and the isotropic hardening σ_h, to the equivalent plastic strain as

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_h(\varepsilon_{pe}) = k(\varepsilon_0 + \varepsilon_{pe})^n$$

here, k is the strength coefficient, n is the hardening exponent, and ε_0 is a reference strain. Noting that at zero plastic strain the initial yield stress is related to the strength coefficient and hardening exponent as

$$\sigma_{ys}(0) = \sigma_{ys0} = k\varepsilon_0^n$$

the yield stress is written as

$$\sigma_{ys}(\varepsilon_{pe}) = k(\varepsilon_0 + \varepsilon_{pe})^n = \sigma_{ys0}\left(1 + \frac{\varepsilon_{pe}}{\varepsilon_0}\right)^n \text{ for } \varepsilon_{pe} \geq 0$$

Voce

The Voce function for nonlinear isotropic hardening is intended for materials that exhibit a saturating evolution of hardening. The isotropic hardening σ_h is exponentially related to the equivalent plastic strain as

$$\sigma_h(\varepsilon_{pe}) = \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{pe}}\right)$$

The yield stress $\sigma_{ys}(\varepsilon_{pe})$ is then defined as

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_h(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{pe}}\right)$$

The value of the *saturation exponent* parameter β determines the saturation rate of the hysteresis loop for cyclic loading. The *saturation flow stress* σ_{sat} characterizes the maximum distance by which the yield surface can expand in the stress space. For values $\varepsilon_{pe} >> 1/\beta$, the yield stress saturates to

$$\sigma_{ys} \to \sigma_{ys0} + \sigma_{sat}$$

Hockett–Sherby

The Hockett–Sherby rule for nonlinear isotropic hardening is also intended for materials which yield stress saturates as equivalent plastic strain increases. It is similar to Voce rule, but it includes an exponential dependency of the form

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + (\sigma_\infty - \sigma_{ys0})\left(1 - e^{-m\varepsilon_{pe}^n}\right)$$

where σ_∞ is the *steady-state flow stress*, m the saturation coefficient and n the saturation exponent. For values $m\varepsilon_{pe}^{\ n} >> 1$ the yield stress saturates to

$$\sigma_{ys} \to \sigma_\infty$$

Hardening Function

The yield stress versus the equivalent plastic strain can be specified with the help of a hardening curve that could also depend on other variables, such as temperature.

In this case, define the (usually nonlinear) hardening function $\sigma_h(\varepsilon_{pe})$ such that the yields stress reads

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_h(\varepsilon_{pe})$$

KINEMATIC HARDENING

A plasticity model with kinematic hardening is characterized by a yield surface that translates during plastic deformation, keeping its initial shape. For example, for the von Mises Criterion adding a kinematic hardening law implies that the radius of the cylinder in stress space is constant, but its axis deviates form the hydrostatic axis.

Many engineering materials, particularly metals, exhibit the *Bauschinger effect*. This phenomenon occurs when a specimen is first loaded in tension until yielding, then

unloaded and reloaded in compression. The compressive yield stress is observed to be lower than if the material had been initially loaded in compression. The Bauschinger effect is commonly explained by kinematic hardening, which is one of the primary motivations for incorporating this hardening mechanism in material models.

A kinematic hardening model can be incorporated into the general yield function in Equation 3-54 by shifting the stress tensor σ by the so-called *back stress* σ_b. The yield function then reads

$$F_y(\sigma - \sigma_b, \sigma_{ys}) = 0$$

or in most cases

$$F_y = \sigma_e(\sigma - \sigma_b) - \sigma_{ys}$$

where σ_e is the scalar equivalent stress. There are three built-in models to define the evolution of the back stress during plastic deformation:

- Linear Hardening
- Armstrong–Frederick Hardening
- Chaboche Hardening

The models are described with the assumption of an additive decomposition of strains and then for a Multiplicative Decomposition.

When kinematic hardening is added, both the plastic potential and the yield function are define by invariants of the effective stress tensor, $\sigma_{eff} = \sigma - \sigma_b$. For example, the second invariant of the effective deviatoric stress tensor is named `solid.II2sEff`.

Linear Hardening

In the linear kinematic hardening model, also known as *Prager's hardening rule*, the evolution the back stress is selected as the hardening variable. The hardening rule is written such that the evolution of the back stress is proportional to the rate of the plastic strain tensor ε_{pl}

$$\dot{\sigma}_b = \frac{2}{3} C_k \dot{\varepsilon}_{pl} \tag{3-75}$$

COMSOL Multiphysics assume that the kinematic hardening modulus C_k is constant, in which case no additional equation is solved, and the back stress σ_b is defined directly from the plastic strain tensor

$$\sigma_b = \frac{2}{3} C_k \varepsilon_{pl}$$

That is, the model does not add additional internal variables. The kinematic hardening modulus C_k is given by

$$C_k = \frac{1}{E_k} - \frac{1}{E}$$

where E_k is the kinematic tangent modulus and the Young's modulus E is taken from the parent material model. If the elasticity of the parent material is anisotropic or orthotropic, E represents an equivalent Young's modulus.

Armstrong–Frederick Hardening

Armstrong and Frederick (Ref. 91) proposed an extension to Prager's linear kinematic hardening model, by introducing additional terms to Equation 3-75

$$\dot{\sigma}_b = \frac{2}{3} C_k \dot{\varepsilon}_{pl} - \lambda \gamma_k \sigma_b \tag{3-76}$$

here, γ_k is a model parameter, C_k is the kinematic hardening modulus, and λ is the plastic multiplier. The additional term introduces an effect of saturation in the hardening rule.

The implementation of the Armstrong–Frederick model in COMSOL Multiphysics uses a back strain tensor ε_b as the hardening variable. The relation between ε_b and the back stress is

$$\sigma_b = \frac{2}{3} C_k \varepsilon_b$$

Considering this relation, Equation 3-76 is rewritten so that the hardening rule to define the evolution of the back strain reads

$$\dot{\varepsilon}_b = \dot{\varepsilon}_{pl} - \lambda \gamma_k \varepsilon_b$$

Chaboche Hardening

Chaboche (Ref. 92) proposed a nonlinear kinematic hardening model based on the superposition of N back stress tensors

$$\sigma_b = \sum_i^N \sigma_{b,i}$$

where each of these back stress tensors $\sigma_{b,i}$ follow a nonlinear Armstrong–Frederick kinematic hardening rule

$$\dot{\sigma}_{b,i} = \frac{2}{3} C_i \dot{\varepsilon}_p - \gamma_i \dot{\varepsilon}_{pe} \sigma_{b,i} \tag{3-77}$$

Practitioners would normally select $\gamma_k = 0$ for one of the back stress equations, thus recovering Prager's linear rule for that branch

$$\sigma_{b,0} = \frac{2}{3} C_0 \varepsilon_p$$

The back stress tensor σ_b is then defined by the superposition of N back stress tensors

$$\sigma_b = \frac{2}{3} C_0 \varepsilon_p + \sum_{i=1}^{N} \sigma_{b,i}$$

As done for the Armstrong–Frederick kinematic hardening, the algorithm solves for the back strain tensors $\varepsilon_{b,i}$ instead of the back stress tensors $\sigma_{b,i}$. The change of variables is

$$\sigma_{b,i} = \frac{2}{3} C_i \varepsilon_{b,i}$$

and the nonlinear evolution for the back strain tensors in Equation 3-77 reads

$$\dot{\varepsilon}_{b,i} = \dot{\varepsilon}_p - \gamma_i \dot{\varepsilon}_{pe} \varepsilon_{b,i}$$

Multiplicative Decomposition

For multiplicative decomposition, the total deformation gradient $\boldsymbol{F}$ is decomposed into an elastic $\boldsymbol{F}_{el}$ and a plastic (inelastic) deformation $\boldsymbol{F}_{pl}$

$$\boldsymbol{F} = \boldsymbol{F}_{el} \boldsymbol{F}_{pl}$$

as described in Inelastic Strain Contributions.

When kinematic hardening is used with multiplicative decomposition, $\boldsymbol{F}_{pl}$ is split as suggested in Ref. 98, such that

$$\boldsymbol{F}_{\mathrm{pl}} = \boldsymbol{F}_{\mathrm{ple}}\boldsymbol{F}_{\mathrm{plv}}$$

Here $\boldsymbol{F}_{\mathrm{ple}}$ is referred to as the "elastic" part of $\boldsymbol{F}_{\mathrm{pl}}$, and $\boldsymbol{F}_{\mathrm{plv}}$ as the "viscous" part of the plastic deformation. Conceptually, $\boldsymbol{F}_{\mathrm{ple}}$ corresponds to a spring and $\boldsymbol{F}_{\mathrm{plv}}$ to a dashpot that are connect in series in a rheological model. Also, new kinematic quantities can be defined in a similar fashion as defined for $\boldsymbol{F}_{\mathrm{el}}$. For example, the tensor

$$\boldsymbol{C}_{\mathrm{ple}} = \boldsymbol{F}_{\mathrm{ple}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{ple}} = \boldsymbol{F}_{\mathrm{plv}}^{-\mathrm{T}}\boldsymbol{C}_{\mathrm{pl}}\boldsymbol{F}_{\mathrm{plv}}^{-1}$$

which is analogous to the elastic right Cauchy–Green deformation tensor $\boldsymbol{C}_{\mathrm{el}}$. As for the elastic part of the model, a free energy function $W_{\mathrm{kin}}(\boldsymbol{C}_{\mathrm{pl,e}})$ is defined for the spring of the kinematic hardening contribution. Different stress tensors can be derived, for example

$$\boldsymbol{S}_{\mathrm{b}} = 2\frac{\partial W_{\mathrm{kin}}(\boldsymbol{C}_{\mathrm{ple}})}{\partial \boldsymbol{C}_{\mathrm{pl}}}$$

which represents that back stress in the material frame and is analogous to the second Piola–Kirchhoff stress of the elastic part. By applying a push-forward operation by the total deformation gradient $\boldsymbol{F}$, the back stress in the spatial frame is

$$\boldsymbol{\sigma}_{\mathrm{b}} = J^{-1}\boldsymbol{F}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{F}^{\mathrm{T}}$$

Alternatively, using the plastic deformation gradient leads

$$\tilde{\boldsymbol{\sigma}}_{\mathrm{b}} = J_{\mathrm{pl}}^{-1}\boldsymbol{F}_{\mathrm{pl}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{F}_{\mathrm{pl}}{}^{\mathrm{T}}$$

where $\tilde{\boldsymbol{\sigma}}_{\mathrm{b}}$ is a representation of the back stress in an intermediate frame defined by $\boldsymbol{F}_{\mathrm{pl}}$. The yield function in Equation 3-67 is defined in the spatial frame for large strains, thus to complete the formulation we need to define $\boldsymbol{\sigma}_{\mathrm{b}}$ or $\boldsymbol{C}_{\mathrm{ple}}$, which depends on the kinematic hardening model used.

For linear elastic materials, it is assumed that W_{kin} has the same form as the elastic strain energy function, and that it is always isotropic. Hence, for a linear elastic material the free energy function W_{kin} reads

$$W_{\mathrm{kin}}(\boldsymbol{\varepsilon}_{\mathrm{ple}}) = \frac{1}{3}C_{\mathrm{k}}[\mathrm{dev}(\boldsymbol{\varepsilon}_{\mathrm{ple}}):\mathrm{dev}(\boldsymbol{\varepsilon}_{\mathrm{ple}})]$$

where $\boldsymbol{\varepsilon}_{\mathrm{ple}} = 0.5(\boldsymbol{C}_{\mathrm{ple}}-1)$ and C_{k} is the kinematic hardening modulus.

For hyperelastic materials, a compressible Neo–Hookean strain energy function is used

$$W_{\text{kin}}(C_{\text{ple}}) = \frac{1}{2}C_{\text{k}}[I_1(C_{\text{ple}}) - 3] - \frac{1}{3}C_{\text{k}}\ln J_{\text{ple}}$$

The flow rule suggested for the Multiplicative Decomposition leads to an arbitrary plastic rotation, which is undesirable with the above formulation for kinematic hardening. Hence, when kinematic hardening is used, the plastic flow rule is modified to

$$\dot{\overline{F_{\text{pl}}^{-1}}} = -\lambda F^{-1}\frac{\partial Q_{\text{p}}}{\partial \sigma}FF_{\text{pl}}^{-1}$$

such that the plastic rotation is uniquely determined.

Linear Hardening To obtain a linear kinematic hardening model the inelastic part of the plastic deformation is omitted by setting $F_{\text{plv}} = 1$, and thus $C_{\text{ple}} = C_{\text{pl}}$.

Armstrong–Frederick Hardening To obtain the Armstrong–Frederick kinematic hardening model, the inelastic part of the plastic deformation F_{plv} is considered as an additional variable determined by the flow rule

$$\dot{\overline{C_{\text{plv}}^{-1}}} = -2\lambda\frac{3\gamma_{\text{k}}}{2C_{\text{k}}}F_{\text{pl}}^{-1}\text{dev}(\tilde{\sigma}_{\text{b}})F_{\text{pl}}C_{\text{plv}}^{-1}$$

where γ_{k} is a kinematic hardening parameter. The actual unknown used by the plasticity algorithm is the symmetric part of the inverse of F_{plv}.

Chaboche Hardening The Chaboche kinematic hardening model can be viewed as a combination of linear kinematic hardening and Armstrong–Frederick kinematic hardening by adding N branches. Formally, each branch corresponds to a unique multiplicative split of F_{pl} such that for branch i the multiplicative decomposition reads

$$F_{\text{pl}} = F_{\text{ple},i}F_{\text{plv},i}$$

Conceptually, each branch corresponds to a spring and a dashpot in series, and all the branches are connected in parallel. The first branch always corresponds to linear kinematic hardening such that $F_{\text{plv},1} = 1$. Each consecutive branch adds an Armstrong–Frederick kinematic hardening model, where $F_{\text{plv},i}$ is an additional variable with its own flow rule. The total back stress is then the sum of the stress in all branches

$$\sigma_b = \sum_i^N \sigma_{b,i}$$

MIXED HARDENING

Both Isotropic Hardening and Kinematic Hardening can be incorporated into the same plasticity model, a formulation known as *mixed hardening*. These more refined hardening models are often necessary to accurately capture the true behavior of many materials.

The general yield function in Equation 3-54 then reads

$$F_y(\sigma - \sigma_b, \sigma_{ys}(\varepsilon_{pe})) = 0$$

or in most cases

$$F_y = \sigma_e(\sigma - \sigma_b) - \sigma_{ys}(\varepsilon_{pe})$$

where σ_e is the scalar equivalent stress.

Porous Plasticity

A key concept for porous plasticity models is the evolution of the *relative density*, which is the solid volume fraction in a porous material. The relative density is related to the porosity (or void volume fraction) ϕ by

$$\rho_{rel} = (1 - \phi)$$

When compacting a mixture of metal particles, the porosity tends to zero and the relative density tends to one. There are different porous plasticity models to account for the mechanism of compaction and void growth.

For a general discussion on plasticity models and their implementation in COMSOL Multiphysics, see The Plasticity Problem.

SHIMA–OYANE CRITERION

Shima and Oyane (Ref. 101) proposed a yield surface for modeling the compaction of porous metallic structures fabricated by sintering. The criterion can be applied for powder compaction at both low and high temperatures.

The yield function and associated plastic potential in the original Shima–Oyane criterion is defined by an ellipsoid in the stress space. The plastic potential Q_p is written in terms of both von Mises equivalent stress and mean pressure, and it also considers isotropic hardening due to changes in porosity. The plastic potential is defined by

$$Q_p(\sigma) = \left(\frac{\sigma_{mises}}{\sigma_{ys}}\right)^2 + \alpha(1-\rho_{rel})^{\gamma}\left(\frac{\sigma_m}{\sigma_{ys}}\right)^2 - \rho_{rel}^m \tag{3-78}$$

where σ_{mises} is the equivalent (von Mises) stress, σ_{ys} is the yield stress, σ_m is the mean stress, and ρ_{rel} is the relative density. The material parameters α, γ, and m are obtained from curve fitting experimental data. Typical material parameter values for copper aggregates are $\alpha = 6.2$, $\gamma = 1.03$, and $m = 5$.

The shape of the yield function in the Octahedral Plane can be made a function of the *Lode angle* θ by using any of the functions described in Octahedral Section.

$$Q_p(\sigma) = \left(\frac{\Gamma(\theta)\sigma_{mises}}{\sigma_{ys}}\right)^2 + \alpha(1-\rho_{rel})^{\gamma}\left(\frac{\sigma_m}{\sigma_{ys}}\right)^2 - \rho_{rel}^m$$

Here, $\Gamma(\theta)$ is a function that defines the shape of the yield function in the Octahedral Plane.

The most common choice, $\Gamma(\theta) = 1$, results in a circular section that recovers the original Shima–Oyane criterion.

GURSON CRITERION

The Gurson criterion (Ref. 102) consists in a pressure-dependent yield function to describe the constitutive response of porous metals. The yield function is derived from the analytical expression of an isolated void immersed in a continuum medium. The void volume fraction, or porosity ϕ, is chosen as main variable.

In Gurson's model, the yield function and associated plastic potential is not an ellipse in the stress space, as in the Shima–Oyane Criterion, but it is defined in terms of the hyperbolic cosine function. The plastic potential for the Gurson criterion reads

$$Q_p(\sigma) = \left(\frac{\sigma_{mises}}{\sigma_{ys}}\right)^2 + 2\phi\cosh\left(\frac{3\sigma_m}{2\sigma_{ys}}\right) - (1+\phi^2) \tag{3-79}$$

here, σ_{mises} is the equivalent (von Mises) stress, σ_{ys} is the yield stress, σ_m is the mean stress, and ϕ is the porosity.

The shape of the yield function in the Octahedral Plane can be made a function of the *Lode angle* θ by using any of the functions described in Octahedral Section.

$$Q_p(\sigma) = \left(\frac{\Gamma(\theta)\sigma_{mises}}{\sigma_{ys}}\right)^2 + 2\phi\cosh\left(\frac{3\sigma_m}{2\sigma_{ys}}\right) - (1+\phi^2)$$

Here, Γ(θ) is a function that defines the shape of the yield function in the Octahedral Plane. The most common choice, Γ(θ) = 1, results in a circular section that recovers the original criterion.

GURSON–TVERGAARD–NEEDLEMAN CRITERION

Tvergaard and Needleman modified the Gurson Criterion for porous plasticity to include parameters to better fit experimental data (Ref. 103-104). The resulting criterion is called the Gurson–Tvergaard–Needleman (GTN) criterion in the literature. The plastic potential for the GTN criterion reads

$$Q_p(\sigma) = \left(\frac{\sigma_{mises}}{\sigma_{ys}}\right)^2 + 2q_1\phi_e\cosh\left(\frac{3q_2\sigma_m}{2\sigma_{ys}}\right) - (1+q_3\phi_e{}^2) \tag{3-80}$$

here, σ_{mises} is the equivalent stress, σ_{ys} is the yield stress, σ_m is the mean stress, and ϕ_e is the effective void volume fraction (effective porosity). Typical correction parameter values are $q_1 = 1.5$, $q_2 = 1.03$, and $q_3 = q_1{}^2$.

The shape of the yield function in the Octahedral Plane can be made a function of the *Lode angle* θ by using any of the functions described in Octahedral Section.

$$Q_p(\sigma) = \left(\frac{\Gamma(\theta)\sigma_{mises}}{\sigma_{ys}}\right)^2 + 2q_1\phi_e\cosh\left(\frac{3q_2\sigma_m}{2\sigma_{ys}}\right) - (1+q_3\phi_e{}^2)$$

Here, Γ(θ) is a function that defines the shape of the yield function in the Octahedral Plane. The most common choice, Γ(θ) = 1, results in a circular section that recovers the original criterion.

The effective void value fraction (or effective porosity) ϕ_e used in the plastic potential is a function of the current porosity ϕ and other material parameters. It is often given by a bilinear function

$$\phi_e = \begin{cases} \phi & \text{if } \phi < \phi_c \\ \phi_c + \dfrac{\phi_m - \phi_c}{\phi_f - \phi_c}(\phi - \phi_c) & \text{if } \phi_c < \phi < \phi_f \\ \phi_m & \text{if } \phi > \phi_f \end{cases}$$

where ϕ_c is the critical void volume fraction (critical porosity) at which void coalescence begins, and ϕ_f is the void volume fraction at failure.

When the porosity increases up to the value of failure, ϕ_f, the effective porosity takes a maximum value of ϕ_m and the porous material loses the capacity to carry stresses. The maximum porosity value is derived from other parameters

$$\phi_m = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3}$$

Since typical values for the parameters are $q_3 = {q_1}^2$, the maximum porosity value is given by $\phi_m = 1/q_1$.

A similar definition for ϕ_e that gives a smooth response as the material reaches failure is suggested in Ref. 106. It is based on a modification such that the effective void volume fraction reaches its maximum value asymptotically

$$\phi_e = \begin{cases} \phi & \text{if } \phi < \phi_c \\ \phi_c + \dfrac{\phi_m - \phi_c}{\phi_f - \phi_c}(\phi - \phi_c) & \text{if } \phi_c < \phi < \phi_i \\ (\phi_m - \phi_s)\left[1 - \exp\left(-\dfrac{\phi - \phi_i}{\phi_f - \phi_i}\right)\right] + \phi_s & \text{if } \phi > \phi_i \end{cases}$$

where

$$\phi_i = \phi_c + \frac{3}{4}(\phi_f - \phi_c)$$

and

$$\phi_s = \phi_c + \frac{\phi_m - \phi_c}{\phi_f - \phi_c}(\phi_i - \phi_c)$$

FLECK–KUHN–MCMEEKING CRITERION

The Fleck–Kuhn–McMeeking (FKM) criterion (Ref. 107), was developed to model the plastic yielding of metal aggregates of high porosity. The yield function and associated plastic potential is derived for randomly distributed particles. The criterion is considered relevant for aggregates with porosity between 10% and 35%. The plastic potential for the FKM criterion reads

$$Q_p = \left(\frac{5}{18}\frac{\sigma_{mises}}{p_f} + \frac{2}{3}\right)^2 + \left(\frac{\sqrt{5}}{3}\frac{\sigma_m}{p_f}\right)^2 - 1 \tag{3-81}$$

here, σ_{mises} is the equivalent (von Mises) stress and σ_m is the mean stress. The *flow strength* of the material under hydrostatic loading, p_f, is computed from

$$p_f = 2.97(1-\phi)^2\frac{\phi_m - \phi}{\phi_m}\sigma_{ys}$$

here, σ_{ys} is the yield stress, and ϕ is the void volume fraction (porosity). The maximum void volume fraction ϕ_m typically takes the value of 36%, the limit of dense random packing of sintered powder.

The shape of the yield function in the Octahedral Plane can be made a function of the *Lode angle* θ by using any of the functions described in Octahedral Section.

$$Q_p = \left(\frac{5}{18}\frac{\Gamma(\theta)\sigma_{mises}}{p_f} + \frac{2}{3}\right)^2 + \left(\frac{\sqrt{5}}{3}\frac{\sigma_m}{p_f}\right)^2 - 1$$

Here, $\Gamma(\theta)$ is a function that defines the shape of the yield function in the Octahedral Plane. The most common choice, $\Gamma(\theta) = 1$, results in a circular section that recovers the original criterion.

The yield function of the original FKM criterion (Ref. 107) is discontinuous at $\sigma_m = 0$. In COMSOL Multiphysics the discontinuity in the plastic potential is resolved as described in Apex Smoothing.

FKM–GTN CRITERION

The FKM–GTN criterion is a combination of the Fleck–Kuhn–McMeeking Criterion (FKM) and the Gurson–Tvergaard–Needleman Criterion (GTN), intended to cover a wider range of porosities (Ref. 108–109). The GTN model is used for low void volume fractions (porosity lower than 10%), and for void volume fractions higher than 25%, the FKM criterion is used. In the transition zone, a linear combination of both criteria

is used. The plastic potential can be associated, or follow either the GTN or FKM potential.

CAPPED DRUCKER–PRAGER

See the sections Drucker–Prager Criterion and Compression Cap for details.

VOID NUCLEATION AND GROWTH

For the porous plasticity criteria, the change in relative density is by default computed from the change in plastic volumetric strain

$$\dot{\rho}_{\text{rel}} = -\rho_{\text{rel}}\dot{\varepsilon}_{\text{p,vol}}$$

Since the relative density is related to the porosity ϕ by $\rho_{\text{rel}} = 1 - \phi$, the change in porosity is also controlled by the change in plastic volumetric strain

$$\dot{\phi} = (1-\phi)\dot{\varepsilon}_{\text{p,vol}}$$

and the change in volumetric plastic strain $\dot{\varepsilon}_{\text{p,vol}}$ is given by the porous plasticity model.

Void nucleation and shear growth are mechanism that can also trigger a change in porosity. The change in porosity based on void nucleation is given by

$$\dot{\phi}_{\text{nucleation}} = \frac{f_N}{s_N\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\varepsilon_{\text{pe}}-\varepsilon_N}{s_N}\right)^2} \dot{\varepsilon}_{\text{pe}} \quad \text{for } p < 0$$

here, ε_N is the mean strain for nucleation, f_N is the void volume fraction for nucleating particles, and s_N is the standard deviation. Typical values for these parameters are $\varepsilon_N = 0.04$, $s_N = 0.1$, and $f_N = 0.04$. It is assumed that nucleation appears only in tension, and that there is no nucleation in compression.

The other mechanism that can affect the porosity is the so-called void growth in shear

$$\dot{\phi}_{\text{shear}} = k_w w \phi \mathbf{n}^D : \dot{\varepsilon}_{\text{pl}}$$

here, k_w is a material parameter, ϕ is the current porosity, $\mathbf{n}^D$ is a deviatoric tensor coaxial to the stress tensor, and $\dot{\varepsilon}_{\text{pl}}$ is the plastic strain rate, which depends on the porous plasticity model. The weight w is computed from the Invariants of the Stress Tensor as

$$w = 1 - \frac{27}{4}\frac{J_3^2}{J_2^3} = \sin^2(3\theta)$$

where θ is the *Lode angle*.

The variable w has a value of $w = 0$ at the compressive and tensile meridians ($\theta = 0$ and $\theta = \pi/3$), and it attains its maximum value, $w = 1$, for $\theta = \pi/6$, which corresponds to maximum shear.

When considering these effects, the change in porosity is controlled by the change in plastic volumetric strain, void nucleation, and shear growth

$$\dot{\phi} = (1 - \phi)\dot{\varepsilon}_{p,vol} + \dot{\phi}_{nucleation} + \dot{\phi}_{shear}$$

ISOTROPIC HARDENING

Porous plasticity models can implement different kinds of isotropic hardening models to describe the hardening of the porous matrix. When applying isotropic hardening, the average flow stress

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\varepsilon_{pe})$$

depends on the initial yield stress σ_{ys0}, the hardening function σ_h, and the equivalent plastic strain in the porous matrix ε_{pe}.

Different isotropic hardening models are implemented for porous plasticity models, which are described for elastoplastic materials in the Isotropic Hardening section.

It is also possible to include a power law relation between the equivalent plastic strain in the porous matrix ε_{pe}, and the flow stress (yield level) σ_{ys}. For uniaxial loading, the strain stress relation is written as (ref. Ref. 104)

$$\varepsilon = \frac{\sigma_{ys0}}{E}\left(\frac{\sigma_{ys}}{\sigma_{ys0}}\right)^m \text{ for } \sigma_{ys} > \sigma_{ys0}$$

where σ_{ys0} is the initial yield stress, m is the stress hardening exponent, and E is the Young's modulus.

By writing the onset of plasticity as $\varepsilon_0 = \sigma_{ys0}/E$, and noting that $\varepsilon = \varepsilon_0 + \varepsilon_{pe}$ for $\sigma_{ys} > \sigma_{ys0}$, this reads

$$\varepsilon_0 + \varepsilon_{pe} = \varepsilon_0\left(\frac{\sigma_{ys}}{\sigma_{ys0}}\right)^m \text{ for } \sigma_{ys} > \sigma_{ys0}$$

which is equivalent to Swift isotropic hardening, when the strain hardening exponent reads $n = 1/m$

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0}\left(1 + \frac{\varepsilon_{pe}}{\varepsilon_0}\right)^n$$

Failure Criteria for Concrete and Rocks

In this section:

- Bresler–Pister Criterion
- Willam–Warnke Criterion
- Ottosen Criterion
- Original Hoek–Brown Criterion
- Generalized Hoek–Brown Criterion

BRESLER–PISTER CRITERION

The Bresler–Pister yield criterion (Ref. 57, Ref. 72) was originally devised to predict the strength of concrete under multiaxial loading. This failure criterion is an extension of the Drucker–Prager Criterion to brittle materials, and it can be expressed in terms of the Invariants of the Stress Tensor as

$$F_y = \sqrt{J_2} + k_1 I_1^2 + k_2 I_1 + k_3 \tag{3-82}$$

here, k_1, k_2, and k_3 are parameters obtained from the uniaxial compressive strength σ_{cs}, the uniaxial tensile strength σ_{ts}, and the biaxial compressive strength σ_{bc}

$$k_1 = \frac{\sigma_{bc}\sigma_{cs} + 2\sigma_{cs}\sigma_{ts} - 3\sigma_{bc}\sigma_{ts}}{\sqrt{3}(\sigma_{cs} + \sigma_{ts})(2\sigma_{bc} - \sigma_{cs})(2\sigma_{bc} + \sigma_{ts})}$$

$$k_2 = \frac{(\sigma_{ts} - \sigma_{cs})(\sigma_{bc}(\sigma_{cs} + \sigma_{ts} - 4\sigma_{bc}) - \sigma_{cs}\sigma_{ts})}{\sqrt{3}(\sigma_{cs} + \sigma_{ts})(2\sigma_{bc} - \sigma_{cs})(2\sigma_{bc} + \sigma_{ts})}$$

$$k_3 = \frac{-\sigma_{cs}\sigma_{bc}\sigma_{ts}(\sigma_{ts} + 8\sigma_{bc} - 3\sigma_{cs})}{\sqrt{3}(\sigma_{cs} + \sigma_{ts})(2\sigma_{bc} - \sigma_{cs})(2\sigma_{bc} + \sigma_{ts})}$$

All the strengths are considered with a positive sign. For typical strength values, the parameters k_1 and k_2 are positive while k_3 is negative.

WILLAM–WARNKE CRITERION

The Willam–Warnke criterion (Ref. 65) is used to predict failure in concrete and other cohesive–frictional materials such as rock, soil, and ceramics. Just as the Bresler–Pister Criterion, it depends only on three parameters. It was developed to describe initial concrete failure under triaxial conditions. The failure surface is convex, continuously differentiable, and it is fitted to test data in the low compression range.

The original "three-parameter" Willam–Warnke failure criterion is defined as

$$F_y = \sqrt{\frac{3}{5}}\frac{\tau_{oct}}{\sigma_{cs}} + r(\theta)\left(\left(\frac{1}{\sigma_{ts}} - \frac{1}{\sigma_{bc}}\right)\sigma_{oct} - 1\right) \tag{3-83}$$

where σ_{cs} is the uniaxial compressive strength, σ_{ts} is uniaxial tensile strength, and σ_{bc} is the biaxial compressive strength (all parameters are positive). The octahedral normal and shear stresses are defined in the Haigh–Westergaard Coordinates

$$\sigma_{oct} = I_1/3 \text{ and } \tau_{oct} = \sqrt{2J_2/3}$$

so the criterion in Equation 3-83 can be written in term of stress invariants as

$$F_y = \sqrt{J_2} + \sqrt{\frac{5}{2}}r(\theta)\sigma_{cs}\left(\left(\frac{1}{\sigma_{ts}} - \frac{1}{\sigma_{bc}}\right)\frac{I_1}{3} - 1\right)$$

The dimensionless function $r(\theta)$ describes the segment of an ellipse on the Octahedral Plane as a function of the Lode angle θ

$$r(\theta) = \frac{2r_c(r_c^2 - r_t^2)\cos\theta + r_c(2r_t - r_c)\sqrt{4(r_c^2 - r_t^2)\cos^2\theta + 5r_t^2 - 4r_c r_t}}{4(r_c^2 - r_t^2)\cos^2\theta + (r_c - 2r_t)^2}$$

Here, the tensile and compressive meridian r_t and r_c are defined in terms of the strengths σ_{cs}, σ_{bc}, and σ_{ts}:

$$r_t = \sqrt{\frac{6}{5}}\ \frac{1}{\sigma_{cs}}\frac{\sigma_{bc}\sigma_{ts}}{2\sigma_{bc} + \sigma_{ts}}$$

$$r_c = \sqrt{\frac{6}{5}}\ \frac{\sigma_{bc}\sigma_{ts}}{3\sigma_{bc}\sigma_{ts} + \sigma_{bc}\sigma_{cs} - \sigma_{ts}\sigma_{cs}}$$

The function $r(\theta)$ can be interpreted as the friction angle which depends on the Lode angle θ (Ref. 65) on the Octahedral Plane.

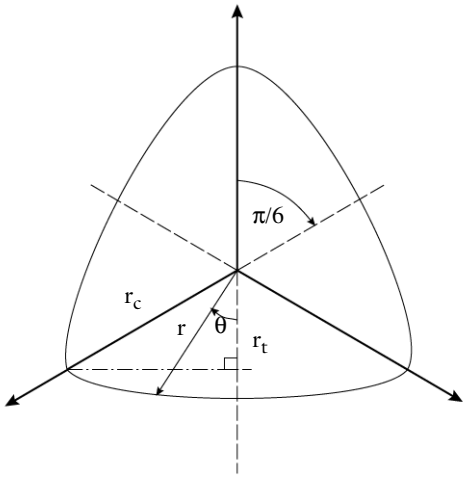


Figure 3-24: The deviatoric section of the Willam–Warnke failure criterion.

OTTOSEN CRITERION

The Ottosen criterion is a four-parameter failure criterion proposed for short-time loading of concrete. It corresponds to a smooth convex failure surface with curved meridians, which is open in the negative (compressive) direction of the hydrostatic axis. The profile on the Octahedral Plane changes from an almost triangular to a more circular shape with increasing hydrostatic pressure. The criterion agrees with experimental results over a wide range of stress states, including both triaxial tests along the tensile and the compressive meridian and biaxial tests (Ref. 73).

The Ottosen criterion is commonly written as (Ref. 72, Ref. 73):

$$F_y = \frac{a}{\sigma_{cs}} J_2 + \lambda(\theta)\sqrt{J_2} + bI_1 - \sigma_{cs}$$

Here, σ_{cs} is the uniaxial compressive strength of concrete, and $a > 0$ and $b > 0$ are dimensionless parameters. The dimensionless function $\lambda(\theta)$ depends on the Lode angle θ and two dimensionless parameters $k_1 > 0$ and $k_2 > 0$.

$$\lambda(\theta) = \begin{cases} k_1 \cos\left(\frac{1}{3}\operatorname{acos}(k_2 \cos(3\theta))\right) & J_3 > 0 \\ k_1 \cos\left(\frac{\pi}{3} - \frac{1}{3}\operatorname{acos}(-k_2 \cos(3\theta))\right) & J_3 < 0, \end{cases}$$

The parameter k_1 is called the *size factor*. The parameter k_2 (also called *shape factor*) is positive and bounded to $0 \le k_2 \le 1$ (Ref. 72, Ref. 73).

Typical values for these parameters are obtained by curve-fitting the uniaxial compressive strength σ_{cs}, uniaxial tensile strength σ_{ts}, and from the biaxial and triaxial data (for instance, a typical biaxial compressive strength of concrete is 16% higher than the uniaxial compressive strength). The parameters σ_{cs}, σ_{bc}, and σ_{ts} are positive.

TABLE 3-4: TYPICAL PARAMETER VALUES FOR OTTOSEN FAILURE CRITERION (Ref. 73).

σ_{ts}/σ_{cs}	a	b	k_1	k_2	λ_t	λ_c
0.08	1.808	4.096	14.486	0.991	14.472	7.783
0.10	1.276	3.196	11.736	0.980	11.711	6.531
0.12	0.922	2.597	9.911	0.965	9.872	5.698

The compressive and tensile meridians (as defined in the Willam–Warnke Criterion) are

$$r_c = \frac{1}{\lambda_c} = \frac{1}{\lambda(\pi/3)} \quad \text{and} \quad r_t = \frac{1}{\lambda_t} = \frac{1}{\lambda(0)}$$

The ratio $r_t/r_c = \lambda_c/\lambda_t$ normally lies between 0.54~0.58 for concrete.

The Ottosen criterion is equivalent to the Drucker–Prager criterion when $a = 0$ and $\lambda(\theta)$ is independent of the Lode angle.

ORIGINAL HOEK–BROWN CRITERION

The Hoek–Brown criterion is an empirical type of model which is commonly used when dealing with rock masses of variable quality. The Hoek–Brown criterion is widely used within civil engineering and is popular because the material parameters can be estimated based on simple field observations together with knowledge of the uniaxial compressive strength of the intact rock material. The Hoek–Brown criterion is one of

the few nonlinear criteria widely accepted and used by engineers to estimate the yield and failure of rock masses. The original Hoek–Brown failure criterion states (Ref. 60)

$$\sigma_1 = \sigma_3 + \sqrt{m\sigma_{cs}\sigma_3 + s\sigma_{cs}^2}$$

where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ are the principal stresses at failure (as defined in geotechnical engineering; that is, an *absolute value*), σ_{cs} is the uniaxial compressive strength of the intact rock, and m and s are positive material parameters.

If the expression is converted into to the sign convention for principal stresses in the Structural Mechanics Module, it becomes

$$\sigma_1 = \sigma_3 + \sqrt{-m\sigma_{cs}\sigma_1 + s\sigma_{cs}^2}$$

with σ_{cs}, m, and s positive material parameters. (In this case, note that $\sigma_1 < s\sigma_{cs}/m$).

As developed originally, there is no relation between the parameters m and s and the physical characteristics of a rock mass measured in laboratory tests. However, for intact rock, $s = 1$ and $m = m_i$, which is measured in a triaxial test.

For jointed rock masses, $0 \leq s < 1$ and $m < m_i$. The parameter m usually lies in the range $5 < m < 30$ (Ref. 62).

TABLE 3-5: CHARACTERISTIC VALUES FOR DIFFERENT ROCK TYPES.

m	ROCK TYPE
5	Carbonate rocks, dolomite, limestone
10	Consolidated rocks, mudstone, shale
15	Sandstone
20	Fine-grained rocks
25	Coarse grained rocks

The Hoek–Brown criterion can be written in terms of the Invariants of the Stress Tensor I_1 and J_2 and the Lode angle $0 \leq \theta \leq \pi/3$, so

$$F_y = 2\sqrt{J_2}\sin\left(\theta + \frac{\pi}{3}\right) - \sigma_{cs}\sqrt{s - m\frac{\sigma_1}{\sigma_c}} = 0$$

GENERALIZED HOEK–BROWN CRITERION

The generalized Hoek–Brown criterion was developed to fit the Geological Strength Index (GSI) classification of isotropic rock masses (Ref. 61). A new relationship between GSI, m, s, and the newly introduced parameter a was developed to give a

smoother transition between very poor quality rock masses (GSI < 25) and stronger rocks

$$\sigma_1 - \sigma_3 = \sigma_{cs}\left(s - m_b\frac{\sigma_1}{\sigma_{cs}}\right)^a$$

where σ_{cs} is the compressive strength of the intact rock. In terms of the invariants J_2 and the Lode angle $0 \le \theta \le \pi/3$ this equals

$$F_y = 2\sqrt{J_2}\sin\left(\theta + \frac{\pi}{3}\right) - \sigma_{cs}\left(s - m_b\frac{\sigma_1}{\sigma_{cs}}\right)^a = 0$$

where $\sigma_1 \ge \sigma_2 \ge \sigma_3$ are the principal stresses (using the Structural Mechanics Module conventions) of the *effective stress tensor* (that is, the stress tensor minus the fluid pore pressure).

The positive parameter m_b is a reduced value of the material constant m_i:

$$m_b = m_i\exp\left(\frac{\mathrm{GSI} - 100}{28 - 14D}\right)$$

s and a are positive parameters for the rock mass given by the following relationships:

$$s = \exp\left(\frac{\mathrm{GSI} - 100}{9 - 3D}\right)$$

$$a = \frac{1}{2} + \frac{1}{6}\left(\exp\left(\frac{-\mathrm{GSI}}{15}\right) - \exp\left(\frac{-20}{3}\right)\right)$$

The *disturbance factor* D was introduced to account for the effects of stress relaxation and blast damage, and it varies from 0 for undisturbed in situ rock masses to 1 for very damaged rock masses.

TABLE 3-6: DISTURBANCE FACTOR IN ROCK MASSES.

D	DESCRIPTION OF ROCK MASS
0	Undisturbed rock mass
0~0.5	Poor quality rock mass
0.8	Damaged rock mass
1.0	Severely damaged rock mass

Damage Models

The deformation of quasi-brittle materials under mechanical loads is characterized by an initial elastic deformation. If a critical level of stress or strain is exceeded, a nonlinear fracture phase will follow the elastic phase.

As this critical value is reached, cracks grow and spread until the material fractures. The occurrence and growth of the cracks play an important role in the failure of brittle materials, and there are a number of theories to describe such behavior.

In the continuum damage mechanics formalism, a *damage* variable represents the amount of deterioration due to crack growth. This damage variable controls the weakening of the material's stiffness, and it produces a nonlinear relation between stress and strain.

For a linear elastic material, Hooke's law relates the *undamaged* stress tensor σ_{un} to the elastic strain tensor:

$$\sigma_{un} = \sigma_{ex} + \mathsf{C} : \varepsilon_{el} = \sigma_{ex} + \mathsf{C} : (\varepsilon - \varepsilon_{inel}) \tag{3-84}$$

here, C is the fourth order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction). The elastic strain ε_{el} is the difference between the total strain ε and inelastic strains ε_{inel}. There may also be an external stress contribution σ_{ex}, with contributions from initial, viscoelastic, or other inelastic stresses.

For Hyperelastic Materials, the undamaged stress tensor follows from the definition of the second Piola–Kirchhoff stress.

For the scalar damage models, the *damaged* stress tensor σ_d is computed from the undamaged stress as

$$\sigma_d = (1 - d)\sigma_{un} \tag{3-85}$$

This damaged stress is then used in the weak formulation. There are different ways to compute the scalar damage variable d that controls the material weakening. These are listed in the following sections. When a large strain formulation is used, σ_{un} and σ_d should be interpreted as the second Piola–Kirchhoff stresses.

The undamaged stress tensor σ_{un} is used when combining the **Damage** feature with Elastoplastic Materials, Elastoplastic Soils, Creep and Viscoplasticity, or Linear Viscoelasticity.

STRAIN-BASED DAMAGE MODELS

The strain-based formulation for the **Scalar damage** and the **Mazars damage for concrete** models is based on the loading function f such as

$$f = \varepsilon_{eq} - \kappa \leq 0 \tag{3-86}$$

here, ε_{eq} is the equivalent strain, a scalar measure of the elastic strain; and κ is a state variable. The evolution of the state variable κ follows the Kuhn–Tucker loading/unloading conditions

$$f \leq 0 \,,\ \dot{\kappa} \geq 0 \,, \text{ and } \dot{\kappa} f = 0$$

In this formulation, κ is the maximum value of ε_{eq} in the load history. The damage variable d is then computed as a function of the state variable κ and other parameters.

Equivalent Strain

Different damage models use different definitions for the equivalent strain ε_{eq}. The **Rankine, stress** damage model defines the equivalent strain from the largest undamaged principal stress σ_{p1} as

$$\varepsilon_{eq} = \frac{\langle \sigma_{p1} \rangle}{E} \tag{3-87}$$

Here, the operator $< \cdot >$ represents the Macaulay brackets and E is Young's modulus. The Macaulay brackets are used since in this formulation only tensile (positive) stresses cause damage.

Similarly, the **Rankine, strain** damage model defines the equivalent strain from the largest principal elastic strain ε_{elp1} as

$$\varepsilon_{eq} = \langle \varepsilon_{elp1} \rangle \tag{3-88}$$

The **Smooth Rankine, stress** damage model defines the equivalent strain from the three undamaged principal stresses

$$\varepsilon_{eq} = \frac{\sqrt{\langle \sigma_{p1} \rangle^2 + \langle \sigma_{p2} \rangle^2 + \langle \sigma_{p3} \rangle^2}}{E} \tag{3-89}$$

In the same way, the **Smooth Rankine, strain** damage model defines the equivalent strain from the three elastic principal strains

$$\varepsilon_{eq} = \sqrt{\langle \varepsilon_{elp1} \rangle^2 + \langle \varepsilon_{elp2} \rangle^2 + \langle \varepsilon_{elp3} \rangle^2} \tag{3-90}$$

By default, only the principal stresses in tension contribute to the damage evolution, for both the Rankine and the Smooth Rankine damage models; but it is also possible to include damage in compression by incorporating the compressive stresses in the computation of the damaged stress tensor σ_d.

The Euclidean **Norm of the elastic strain tensor** can also be used as a measure for the equivalent strain

$$\varepsilon_{eq} = \sqrt{\varepsilon_{el} : \varepsilon_{el}} \tag{3-91}$$

The Euclidean norm considers both tensile and compressive strains.

For the **Mazars damage for concrete** model, it is also possible to select from **Mazars** or **Modified Mazars** equivalent strain. Mazars equivalent strain is defined as

$$\varepsilon_{eq} = \sqrt{\langle \varepsilon_{el} \rangle : \langle \varepsilon_{el} \rangle} \tag{3-92}$$

In the modified Mazars equivalent strain (Ref. 121 and Ref. 122), a correction factor γ is added to improve the approximation of the failure surface of concrete in multiaxial compression.

$$\varepsilon_{eq} = \gamma \sqrt{\langle \varepsilon_{el} \rangle : \langle \varepsilon_{el} \rangle} \tag{3-93}$$

It is also possible to use a **User defined** expression for defining the equivalent strain as a function of undamaged stress, stress components, or strains.

Damage Evolution

A key component in a scalar damage model is the definition of the damage evolution law. The **Linear strain softening** law defines the damage variable from

$$\begin{aligned} d(\kappa) &= \left(1 - \frac{\varepsilon_0}{\kappa}\right)\left(\frac{\varepsilon_f}{\varepsilon_f - \varepsilon_0}\right) \quad && \kappa \geq \varepsilon_0 \\ d(\kappa) &= 0 && \kappa < \varepsilon_0 \end{aligned}, \tag{3-94}$$

Here, ε_0 denotes the onset of damage, computed from the tensile strength σ_{ts} and Young's modulus E, so that $\varepsilon_0 = \sigma_{ts}/E$. The parameter ε_f is derived from other material parameters such as the tensile strength, the characteristic element size h_{cb} and the fracture energy per reference area G_f, or the fracture energy per reference volume g_f.

$$\varepsilon_f = \frac{2G_f}{\sigma_{ts} h_{cb}} + \frac{\varepsilon_0}{2} \tag{3-95}$$

The **Exponential strain softening** law defines the damage evolution from

$$d(\kappa) = 1 - \frac{\varepsilon_0}{\kappa}\exp\left(-\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0}\right) \qquad \kappa \geq \varepsilon_0$$
$$d(\kappa) = 0 \qquad \kappa < \varepsilon_0 \tag{3-96}$$

where

$$\varepsilon_f = \frac{G_f}{\sigma_{ts} h_{cb}} + \frac{\varepsilon_0}{2} \tag{3-97}$$

The **Polynomial strain softening** law defines the damage evolutions from

$$d(\kappa) = 1 - \frac{\varepsilon_0}{\kappa}\left[1 + \left(\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0}\right)^2\left(2\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0} - 3\right)\right] \qquad \kappa \geq \varepsilon_0$$
$$d(\kappa) = 0 \qquad \kappa < \varepsilon_0 \tag{3-98}$$

with ε_f according to Equation 3-95.

The **Multilinear strain softening** law defines the damage evolutions from

$$d(\kappa) = 1 - \frac{\varepsilon_0}{\kappa}\frac{\varepsilon_f - \kappa}{\varepsilon_f - \varepsilon_p} \qquad \kappa \geq \varepsilon_p$$
$$d(\kappa) = 1 - \frac{\varepsilon_0}{\kappa} \qquad \varepsilon_0 < \kappa < \varepsilon_p$$
$$d(\kappa) = 0 \qquad \kappa < \varepsilon_0 \tag{3-99}$$

with ε_f according to Equation 3-95 and

$$\varepsilon_p = \lambda\frac{G_f}{\sigma_{ts} h_{cb}} + \varepsilon_0\left(1 - \frac{\lambda}{2}\right) \tag{3-100}$$

where λ is an input parameter that defines the shape of the curve.

Mazars Damage for Concrete

The **Mazars damage for concrete** utilizes two different damage evolution laws, one for tensile damage and another for compressive damage. These two damage functions are combined as

$$d(\kappa) = \alpha_t^\beta d_t(\kappa) + \alpha_c^\beta d_c(\kappa) \tag{3-101}$$

here, α_t and α_c are weight functions depending on the current stress state, and β is the so-called *shear exponent*, which determines the response in shear, that is, the evolution of the combined damage function in states where both damage functions are active.

To define the tensile damage evolution law $d_t(\kappa)$ (Ref. 122), it is possible to use either **Linear strain softening**, **Exponential strain softening**, described in Damage Evolution section; or **Mazars damage evolution function**. A tensile *Mazars damage evolution function* is obtained by a combination of linear and exponential strain softening

$$\begin{aligned} d_t(\kappa) &= 1-(1-A_t)\frac{\varepsilon_{0t}}{\kappa}-A_t\exp(-B_t(\kappa-\varepsilon_{0t})) && \kappa \geq \varepsilon_{0t} \\ d_t(\kappa) &= 0 && \kappa < \varepsilon_{0t} \end{aligned} \tag{3-102}$$

Here, A_t and B_t are tensile damage evolution parameters, and ε_{0t} is the tensile strain threshold.

The compressive *Mazars damage evolution function* $d_c(\kappa)$ is obtained by

$$\begin{aligned} d_c(\kappa) &= 1-(1-A_c)\frac{\varepsilon_{0c}}{\kappa}-A_c\exp(-B_c(\kappa-\varepsilon_{0c})) && \kappa \geq \varepsilon_{0c} \\ d_c(\kappa) &= 0 && \kappa < \varepsilon_{0c} \end{aligned} \tag{3-103}$$

Here, A_c and B_c are compressive damage evolution parameters, and ε_{0c} is the compressive strain threshold. Both the tensile and compressive damage evolution laws can also be specified by **User defined** expressions.

Spatial Regularization

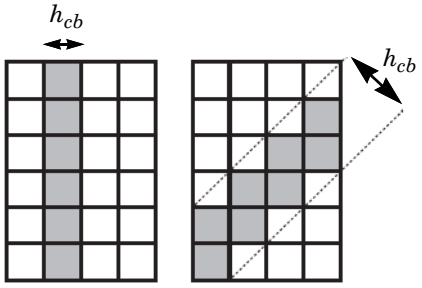
The most common application for the damage models is to describe strain localization, due to cracking in quasi-brittle materials. In a damage model without regularization, the deformation during strain softening will always localize in the narrowest possible band, following the principle of least action. This means that large strains will develop in a narrow band of elements (or even Gauss points). As a consequence, the amount of energy dissipated during softening will decrease upon mesh refinement. The results of a damage model without regularization will therefore be mesh dependent and possibly unstable; hence the need for these models to be *regularized* for the solution to maintain its mesh objectivity.

The simplest regularization method is to modify the stress–strain relation to account for the mesh size. More advanced regularization techniques introduce length scales in the constitutive equation, and additional equations and variables acting as *localization*

limiters. The following sections describe the two methods for damage regularization available in COMSOL Multiphysics.

The Crack Band Method

This simplest regularization technique is based on stress–strain curves (damage evolution laws) that depend on the mesh and element characteristics. The method is often called the *Crack Band* method (Ref. 123, Ref. 124). The method regularizes the solution from a global viewpoint, which dissipates the correct amount of energy during strain localization. The main difficulty when using the crack band method is to find the correct width of the crack band, h_{cb}; which can depend on the element size and shape, as well as the order of the interpolation and the current stress state (that is, the inclination of the crack with respect to the mesh).



The length scale used in the crack band method is computed by using the volume to area ratio in the mesh elements. By using the element volume v in 3D or the area a in 2D, the crack bandwidth h_{cb} is defined as

- $h_{\text{cb}} = \sqrt{2a}$, for 2D triangles
- $h_{\text{cb}} = \sqrt{a}$, for 2D rectangles
- $h_{\text{cb}} = \sqrt[3]{6\sqrt{2}v}$, for 3D tetrahedra
- $h_{\text{cb}} = \sqrt[3]{6v/\sqrt{2}}$, for 3D pyramids
- $h_{\text{cb}} = \sqrt[3]{2v}$, for 3D wedges
- $h_{\text{cb}} = \sqrt[3]{v}$, for 3D hexahedra

The crack band width h_{cb} is then used to modify the Damage Evolution law in which the damage variable $d(\kappa)$ is computed. Note that the damage evolution laws (Equation 3-102 and Equation 3-103) are unaffected by the crack band method.

The Implicit Gradient Method

The *Implicit Gradient* method (Ref. 125) enforces a predefined width of the damage zone through a localization limiter. This is achieved by adding a nonlocal strain variable, the *nonlocal equivalent strain* ε_{nl}, through an additional PDE where the equivalent strain ε_{eq} acts as source term. This PDE is solved simultaneously with the displacement field:

$$\varepsilon_{nl} - c\nabla^2\varepsilon_{nl} = \varepsilon_{eq} \tag{3-104}$$

Here, the parameter c controls the width of the localization band. This parameter is defined from the internal length scale, l_{int}, and the geometry dimension n (two or three dimension)

$$c = \frac{l_{int}^2}{2n} \tag{3-105}$$

If the fracture energy per reference area is used to define the softening behavior, the size of the damage zone is also needed as input. This parameter does not necessarily have the same value as l_{int}.

The strain-based formulation for the damage model (Equation 3-86) is then redefined by the nonlocal equivalent strain ε_{nl} instead of the equivalent strain ε_{eq}

$$f = \varepsilon_{nl} - \kappa \leq 0 \tag{3-106}$$

Viscous Regularization

A rate dependency can be introduced to the damage model by using the **Delayed damage** viscous regularization method available for time-dependent studies. The method can be used to model a real rate-dependent fracture problem, as well as for stabilizing a rate-independent problem. In both cases, a viscous damage variable d_v is introduced. This variable is defined through the following ODE

$$\tau\frac{\partial d_v}{\partial t} + d_v = d(\kappa) \tag{3-107}$$

where τ is the characteristic relaxation time. For the case when the viscous regularization is used to stabilize a rate-independent fracture problem, τ should be significantly smaller than the time step used by the solver in order not to modify the properties of the problem too much. The viscous damage variable d_v is then used in Equation 3-85 to define the damaged stress tensor.

PHASE-FIELD DAMAGE MODELS

The **Phase-field damage** model is closely related to the Strain-based Damage Models, but it is derived from a different starting point. Following a similar derivation as reported in Ref. 126, the model is based on the regularization of a variational formulation of the classical theory by Griffith for brittle fracture where the geometric crack discontinuity is regularized by a second-order phase field. The energy functional governing the crack propagation takes the following form:

$$E(\mathbf{u}, \nabla\mathbf{u}, \phi, \nabla\phi) = \int_{\Omega} (1 - d(\phi)) W_{s0} d\mathbf{x} + \int_{\Omega} G_c \left(\frac{1}{2 l_{int}} \phi^2 + \frac{l_{int}}{2} |\nabla\phi|^2 \right) d\mathbf{x} \quad (3\text{-}108)$$

where ϕ is the crack phase field, $d(\phi)$ is the damage function, W_{s0} is the undamaged elastic strain energy density, G_c is the critical energy release rate, and l_{int} is the internal length scale that appears from the regularization of the discrete crack.

Given the two dependent variables, $\mathbf{u}$ and ϕ, the main contributions to the coupled initial boundary value problem (including inertial terms), are:

$$\begin{cases} \rho \dfrac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \sigma_d + \mathbf{f}_V & \text{in } \Omega \\ \tau \dfrac{\partial \phi}{\partial t} = \dfrac{\partial d(\phi)}{\partial \phi} H_d - (\phi - l_{int}^2 \nabla^2 \phi) & \text{in } \Omega \\ 0 = -\nabla\phi \cdot \mathbf{n} & \text{on } \partial\Omega \end{cases} \quad (3\text{-}109)$$

where a viscous regularization term with the characteristic time τ has been included and the state variable H_d have been introduced. The state variable H_d is a function of the crack driving force D_d, and it satisfies the Kuhn–Tucker conditions so that:

$$H_d(t, \mathbf{u}) = \max_{\tau \in [0, t]} D_d(\tau, \mathbf{u}) \quad (3\text{-}110)$$

The momentum balance in Equation 3-109 is included for completeness but is not implemented directly as part of the phase-field damage model. However, the equation gives an indication that the damaged stress, as defined by Equation 3-85, is used for the mechanical equilibrium.

Crack Driving Force

The crack driving force, D_d, can be either a function of the elastic strain energy density, or it can also include dissipation processes.

By selecting the **Elastic strain energy density**, the crack driving force is derived from

$$D_{\mathrm{d}} = \frac{l_{\mathrm{int}}}{G_{\mathrm{c}} - G_{\mathrm{c0}}} \langle W_{\mathrm{s0}}^{+} - \frac{G_{\mathrm{c0}}}{l_{\mathrm{int}}} \rangle \tag{3-111}$$

where $W_{\mathrm{s0}}{}^{+}$ is the tensile part of the undamaged elastic strain energy density. The strain energy threshold G_{c0} is an ad hoc variable introduced to establish an elastic threshold before damage occurs. Setting G_{c0} equal to zero recovers the model reported in Ref. 126.

By selecting **Total strain energy density**, additional contributions from dissipative material features, such as creep, plasticity, and viscoelasticity, are added. Select the **Calculate energy dissipation** checkbox in the parent material model to compute these variables.

An alternative definition for the crack driving force based on stress measures is suggested in Ref. 128. By selecting the **Principal stress criterion**, the crack driving force is defined as

$$D_{\mathrm{d}} = \xi \langle \sum_{i=1}^{3} \left(\frac{\sigma_{\mathrm{p}i}}{\sigma_{\mathrm{cr}}} \right)^2 \rangle \tag{3-112}$$

where σ_{cr} is the *critical fracture stress*, $\sigma_{\mathrm{p}i}$ are the *undamaged principal stresses*, and ξ is a dimensionless parameter which determines the post peak slope.

Another stress-based definition of the crack driving force is the **Rankine criterion**, for which the crack driving force is defined as (Ref. 129)

$$D_{\mathrm{d}} = \frac{l_{\mathrm{int}}}{2G_{\mathrm{c}}E} \langle \sigma_{\mathrm{p1}} \rangle^2 \tag{3-113}$$

It is also possible to enter a **User defined** expression for the crack driving force D_{d}.

Damage Evolution for Phase-Field Damage

The damage variable d is in the phase-field damage model a function of the crack phase field ϕ. By selecting **Power law**, the damage evolution function is defined as

$$d(\phi) = 1 - (1 - \phi)^m \tag{3-114}$$

where the exponent m is a user input. Often a quadratic function is used so that $m = 2$.

An alternative damage function was suggested in Ref. 130. By selecting **Cubic, Borden**, the damage evolution function $d(\phi)$ reads

$$d(\phi) = 1 - s[(1-\phi)^3 - (1-\phi)^2] - 3(1-\phi)^2 + 2(1-\phi)^3 \quad (3\text{-}115)$$

where the model parameter s controls the shape of the damage evolution function. The **Cubic, Borden** option is only available when the crack driving force is defined through the strain energy density.

A general form of the damage evolution law based on a **Rational** function of the phase field was suggested in Ref. 129. This type of damage evolution function is defined as

$$d(\phi) = \frac{Q(\phi)}{(1-\phi)^p + Q(\phi)}, Q(\phi) = a_1\phi(1 + a_2\phi(1 + a_3\phi)) \quad (3\text{-}116)$$

with the model parameters p, a_1, a_2, and a_3. The value of the exponent p should be greater than or equal to 2.

The options **Linear softening**, **Exponential softening**, and **Cornelissen softening** are special cases of the **Rational** damage evolution function intended for modeling quasi-brittle fracture inspired by cohesive zone models, see Figure 3-25. In these cases, the parameters p, a_2, and a_3 are predetermined constants, and a_1 is computed internally from the **Critical stress** σ_{cr}, critical energy release rate G_c, equivalent elastic modulus E_{equ}, and internal length scale l_{int} as $a_1 = 4G_cE_{equ}/(\pi\sigma_{cr}^2 l_{int})$.

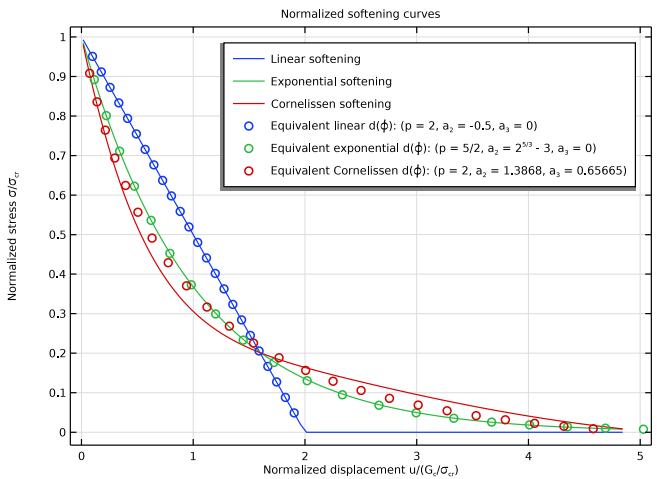


Figure 3-25: Equivalent damage evolution functions (circles) for the linear, exponential, and Cornelissen softening laws (lines).

It is also possible to enter a **User defined** expression for the damage evolution d as a function of the phase field variable ϕ.

Strain Energy Split

Typically, phase-field damage models use a decomposition of the undamaged strain energy density W_{s0} into a tensile and compressive parts. Only the tensile part W_{s0}^{+} affects the phase field, and consequently the damage growth and fracture. Several alternative definitions of W_{s0}^{+} are available in COMSOL Multiphysics.

The simplest alternative is to set W_{s0}^{+} equal to the complete strain energy density W_{s0}, which means that no split is made into tensile and compressive counterparts. Select **No split** in the **Exclude compressive energy** list to achieve this. This option has the advantage that it greatly simplifies the structure of the equations, although that comes at the cost of physical correctness, especially for shear and cyclic loading.

A more accurate method is to use the volumetric-deviatoric split by selecting **Volumetric only** in the **Exclude compressive energy** list. Here, only the positive part of the volumetric stress is considered while the entire undamaged deviatoric stress $\sigma_{dev} = \text{dev}(\sigma_{un})$ is included in W_{s0}^{+}. The definition of W_{s0}^{+} then reads

$$W_{s0}^{+} = \frac{1}{2}\left(\frac{\langle I_1 \rangle}{3}\varepsilon_{el,vol} + \sigma_{dev}:\varepsilon_{el,dev}\right) \tag{3-117}$$

where $I_1 = \text{trace}(\sigma_{un})$ is the first invariant of the undamaged stress tensor defined in Equation 3-84 (see Invariants of the Stress Tensor); and $\varepsilon_{el,vol}$ and $\varepsilon_{el,dev}$ are the volumetric and deviatoric parts of the elastic strain tensor, respectively (see Invariants of Strain). For hyperelastic materials, Equation 3-117 is replaced by

$$W_{s0}^{+} = \begin{cases} W_{iso} + W_{vol} & \text{if } J_{el} \geq 1 \\ W_{iso} & \text{if } J_{el} < 1 \end{cases}$$

where W_{iso} is the isochoric strain energy density, W_{vol} is the volumetric strain energy density, and J_{el} is the elastic volume ratio. The split is only available if the parent hyperelastic model permits a clear split of W_s into two parts.

Another alternative is to define W_{s0}^{+} by performing a spectral decomposition of the stress tensor such that

$$W_{s0}^{+} = \frac{1}{2}\sum_{i=1}^{3} \langle \sigma_{pi} \rangle \varepsilon_{el,\,pi} \tag{3-118}$$

where σ_{pi} are the principal values of the undamaged stress tensor σ_{un}, and $\varepsilon_{el,pi}$ are the principal values of the elastic strain tensor ε_{el} (see Principal Strains).

Similarly, a spectral decomposition can be made directly on the elastic strain tensor as suggested in Ref. 126. The tensile part of the elastic strain tensor is then

$$\varepsilon_{el}^{+} = \sum_{i=1}^{3} \langle \varepsilon_{el,\,pi} \rangle \mathbf{n}_i \otimes \mathbf{n}_i \tag{3-119}$$

and the tensile part of the undamaged stress tensor is defined as

$$\sigma^{+} = C : \varepsilon_{el}^{+} \tag{3-120}$$

Using these two tensors, the tensile part of the elastic strain energy density is

$$W_{s0}^{+} = \frac{1}{2}\sigma^{+} : \varepsilon_{el}^{+} \tag{3-121}$$

Both the **Spectral decomposition, stress** and **Spectral decomposition, strain** options involve computing the principal values and principal vectors. Such operations are comparatively expensive and typically increase the computational cost of the phase-field damage method; however, these are the most physically correct formulations to use.

The choice of how to split the elastic strain energy density also affects how the damaged stress tensor is constructed form the tensile part of the undamaged stress tensor.

Coupled Damage–Plasticity

The coupled damage–plasticity model combines damage mechanics with a plasticity model to describe important characteristics of the mechanical response and failure of concrete and similar materials subjected to multiaxial and cyclic loading conditions.

Specifically, it implements the constitutive model for concrete suggested in Ref. 82 and Ref. 83.

The behavior of the coupled damage–plasticity model when subjected to different loading conditions is exemplified in *Concrete Damage–Plasticity Material Tests*: Application Library path **Geomechanics_Module/Verification_Examples/concrete_damage_plasticity**.

GENERAL EQUATIONS

The general constitutive equations for the coupled damage–plasticity model uses the concept of the *damaged* stress tensor σ_d and the *undamaged* stress tensor σ_{un}. The former is used in the weak formulation while the latter describe the behavior of the intact material.

The damaged stress is computed from the following constitutive equation

$$\sigma_d = (1-d_t)\sigma_{un}^{+} + (1-d_c)\sigma_{un}^{-}$$

where σ_{un} is split into a positive and negative part by a spectral decomposition in order to account for the anisotropic behavior of concrete in tension and compression. For the same reason, two damage variables introduced: the tensile damage variable d_t and the compressive damage variable d_c. These variables introduce a weakening of the material's stiffness, and varies from 0 for an undamaged state to 1 for a fully damaged state. Their definitions are described in Damage Model.

Assuming an additive decomposition of strains, the behavior of the intact material is commonly described by a linear elastic material, such that Hooke's law relates σ_{un} to the elastic strain tensor

$$\sigma_{un} = \sigma_{ex} + C : \varepsilon_{el} = \sigma_{ex} + C : (\varepsilon - \varepsilon_{inel})$$

where, C is the fourth order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction). The elastic strain ε_{el} is the difference between the total strain ε and inelastic strains ε_{inel}. The coupled damage–plasticity model always adds a plastic strain tensor ε_{pl} to ε_{inel}, but other contributions such as thermal strains can also be considered. There may also be an external stress contribution σ_{ex}, with contributions from initial, viscoelastic, or other inelastic stresses. See also Inelastic Strain Contributions.

PLASTICITY MODEL

The evolution of the plastic deformation is based on the undamaged stress, and thus is independent of damage. A flow plasticity model with isotropic hardening is used , it introduces a yield function F_p, a plastic potential Q_p, a scalar hardening variable κ_p, and the plastic multiplier λ_p. See also The Plasticity Problem for a general description of how plasticity is solved in COMSOL Multiphysics.

The model can be summarized by the following evolution equations for the plastic strains ε_{pl} and the hardening variable κ_p

$$\dot{\varepsilon}_{pl} = \dot{\lambda}_p \frac{\partial Q_p(\sigma_{un}, \kappa_p)}{\partial \sigma_{un}}$$
$$\dot{\kappa}_p = h_p(\sigma_{un}, \varepsilon_{pl})$$

The evolution equations are solved together with the Kuhn–Tucker conditions

$$F_p \leq 0, \dot{\lambda}_p \geq 0, \dot{\lambda}_p F_p = 0 \ .$$

See the general description of a Multiplicative Plastic Flow rule for large plastic strains.

The yield function uses a generalization of Willam–Warnke Criterion written as a function of the hardening variable and κ_p, and the undamaged stress given in the Haigh–Westergaard Coordinates (ξ_{un}, r_{un}, θ_{un})

The yield function is defined as

$$F_p(\zeta_{un}, r_{un}, \theta_{un}, \kappa_p) = \left\{ [1 - q_{h1}(\kappa_p)] \left(\frac{r_{un}}{\sqrt{6}\sigma_{cs}} + \frac{\zeta_{un}}{\sigma_{cs}} \right)^2 + \sqrt{\frac{3}{2}} \frac{r_{un}}{\sigma_{cs}} \right\}^2 \qquad (3\text{-}122)$$
$$+ m_0 q_{h1}^2(\kappa_p) q_{h2}(\kappa_p) \left[\frac{r_{un}}{\sqrt{6}\sigma_{cs}} r(\theta_{un}) + \frac{\zeta_{un}}{\sigma_{cs}} \right] + q_{h1}^2(\kappa_p) q_{h2}(\kappa_p)$$

here, $q_{h1}(\kappa_p)$ and $q_{h2}(\kappa_p)$ are hardening functions, and σ_{cs} is the *uniaxial compressive strength*. The friction parameter m_0 is written as

$$m_0 = \frac{3(\sigma_{cs}^2 - \sigma_{ts}^2)}{\sigma_{cs}\sigma_{ts}} \frac{e}{e+1}$$

where σ_{ts} is the *uniaxial tensile strength* and e is the *eccentricity* parameter.

The meridians of the yield surface F_p on the Meridional Plane are parabolic, while the section on the Octahedral Plane is triangular at low confinement and changes toward a circular shape at high confinement. The shape of the section is given by a dimensionless function of the undamaged Lode angle θ_{un}

$$r(\theta_{un}) = \frac{4(1-e^2)\cos^2\theta_{un} + (2e-1)^2}{2(1-e^2)\cos\theta_{un} + (2e-1)\sqrt{4(1-e^2)\cos^2\theta_{un} + 5e^2 - 4e}}$$

As suggested in Ref. 82 and Ref. 83, the eccentricity is computed as

$$e = \frac{1+p}{2-p}, \text{ with } p = \frac{\sigma_{ts}}{\sigma_{bc}}\frac{\sigma_{bc}^2 - \sigma_{cs}^2}{\sigma_{bc}^2 - \sigma_{ts}^2}$$

where σ_{bc} is the *biaxial compressive strength*.

A graphical representation of yield surface $F_p = 0$ in the undamaged principal stress space and its evolution after hardening is shown in Figure 3-26 and Figure 3-27.

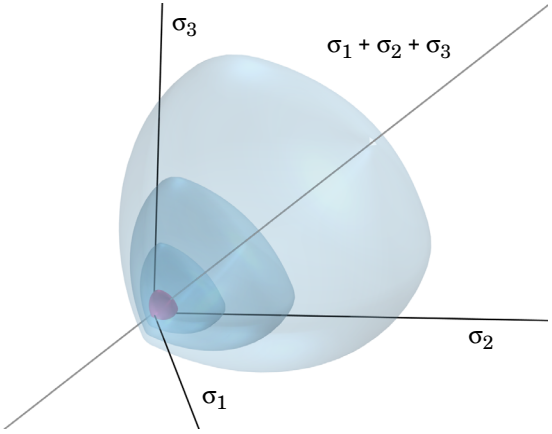


Figure 3-26: Graphical representation of the yield surface in the undamaged principal stress space at different hardening levels ($\kappa_p < 0.7$). The red surface shows the initial yield surface at $\kappa_p = 0$.

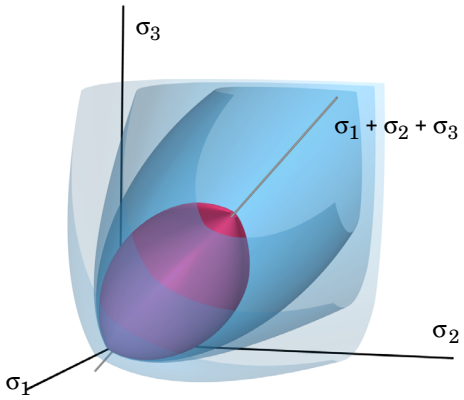


Figure 3-27: Graphical representation of the yield surface in the undamaged principal stress space at different hardening levels ($\kappa_p > 0.7$).

Plastic Flow

The plastic flow is determined by a nonassociated flow rule, since an associated flow rule typically leads to an overestimation of the concrete strength due to an unrealistically large volume expansion. Here the plastic potential is a simplified version of the yield function in that it is independent of the undamaged Lode angle θ_{un},

$$Q_p(\zeta_{un}, r_{un}, \kappa_p) = \left\{[1 - q_{h1}(\kappa_p)]\left(\frac{r_{un}}{\sqrt{6}\sigma_{cs}} + \frac{\zeta_{un}}{\sigma_{cs}}\right)^2 + \sqrt{\frac{3}{2}}\frac{r_{un}}{\sigma_{cs}}\right\}^2 \qquad (3\text{-}123)$$
$$+ q_{h1}^2(\kappa_p)\left[\frac{m_0 r_{un}}{\sqrt{6}\sigma_{cs}} + \frac{m_q(\zeta_{un}, \kappa_p)}{\sigma_{cs}}\right]$$

with

$$m_q(\zeta_{un}, \kappa_p) = A_q(\kappa_p) B_q(\kappa_p) \sigma_{cs} \exp \frac{\zeta_{un} - q_{h2}(\kappa_p)\frac{\sigma_{ts}}{3}}{B(\kappa_p)\sigma_{cs}}$$

where $A_q(\kappa_p)$ and $B_q(\kappa_p)$ are auxiliary functions derived from the plastic flow in the post-peak regime, see Ref. 83. The functions are defined as

$$A_q(\kappa_p) = \frac{\sigma_{ts} q_{h2}(\kappa_p)}{\sigma_{cs}} + \frac{m_0}{2}$$

and

$$B_q(\kappa_p) = \frac{\frac{q_{h2}(\kappa_p)}{3}\left(1+\frac{\sigma_{ts}}{\sigma_{cs}}\right)}{\ln A_q(\kappa_p) + \ln(2D_f - 1) - \ln\left[3q_{h2}(\kappa_p) + \frac{m_0}{2}\right] + \ln(D_f + 1)}$$

where D_f is a model parameter that controls the dilatancy of the plastic flow. By default its value reads $D_f = 0.85$.

The plastic potential in Equation 3-123 leads to a singularity at the point where the plastic potential intersects the hydrostatic axis. Such formulations are not suitable for a robust numerical implementation. To overcome this, COMSOL Multiphysics modifies Equation 3-123 such that

$$Q_p = \left\{[1 - q_{h1}(\kappa_p)]\left(\frac{\sqrt{2J_2 + \sigma_{v,off}^2 m_0^2}}{\sqrt{6}\sigma_{cs}} + \frac{\zeta_{un}}{\sigma_{cs}}\right)^2 + \sqrt{\frac{3}{2}}\frac{\sqrt{2J_2 + \sigma_{v,off}^2 m_0^2}}{\sigma_{cs}}\right\}^2$$
$$+ q_{h1}^2(\kappa_p)\left[\frac{m_0\sqrt{2J_2 + \sigma_{v,off}^2 m_0^2}}{\sqrt{6}\sigma_{cs}} + \frac{m_q(\zeta_{un}, \kappa_p)}{\sigma_{cs}}\right]$$

where J_2 is the second invariant of the undamaged stress tensor. The smoothing parameter $\sigma_{v,off}$ controls the distance between where the potential in Equation 3-123 and the smooth potential intersects the hydrostatic axis as shown in Figure 3-18 for Drucker–Prager criterion. The parameter $\sigma_{v,off}$ can be estimated as a fraction of the uniaxial compressive strength σ_{cs}. See Apex Smoothing for details.

Hardening

Isotropic hardening controls the evolution of the yield surface during plastic flow. The evolution of the scalar hardening variable κ_p is proportional to the norm of the plastic strain tensor and follows the evolution equation

$$\dot{\kappa}_p = \frac{\|\varepsilon_{pl}\|}{x_h(\zeta_{un})}(2\cos\theta_{un})^2$$

The rate of κ_p depends on the direction of loading such that it grows faster in uniaxial compression, while it is constant in uniaxial tension. The rate of κ_p is, furthermore, scaled by a *ductility* measure

$$x_{\mathrm{h}}(\zeta_{\mathrm{un}}) = \begin{cases} A_{\mathrm{h}} + (A_{\mathrm{h}} - B_{\mathrm{h}})\exp\left[-\dfrac{R_{\mathrm{h}}(\zeta_{\mathrm{un}})}{C_{\mathrm{h}}}\right] & \text{if} \quad R_{\mathrm{h}}(\zeta_{\mathrm{un}}) \geq 0 \\ (B_{\mathrm{h}} - D_{\mathrm{h}})\exp\left[\dfrac{(A_{\mathrm{h}} - B_{\mathrm{h}})R_{\mathrm{h}}(\zeta_{\mathrm{un}})}{(B_{\mathrm{h}} - D_{\mathrm{h}})C_{\mathrm{h}}}\right] + D_{\mathrm{h}} & \text{otherwise} \end{cases}$$

such that the material is more ductile in compression. Here A_{h}, B_{h}, C_{h}, and D_{h} are model parameters, and

$$R_{\mathrm{h}}(\zeta_{\mathrm{un}}) = -\frac{\zeta_{\mathrm{un}}}{\sigma_{\mathrm{cs}}} - \frac{1}{3}$$

COMSOL Multiphysics uses default suggestions for these parameters according to the suggestions given in Ref. 82, where also a calibration procedure from measured data is outlined.

The hardening variable κ_{p} enters the two hardening functions q_{h1} and q_{h2} that appear in Equation 3-122 and Equation 3-123. These functions control the evolution of the size and shape of the yield function and the plastic potential. They are defined as

$$q_{\mathrm{h1}}(\kappa_{\mathrm{p}}) = \begin{cases} q_{\mathrm{h0}} + (1 - q_{\mathrm{h0}})(\kappa_{\mathrm{p}}^3 - 3\kappa_{\mathrm{p}}^2 + 3\kappa_{\mathrm{p}}) - H_{\mathrm{p}}(\kappa_{\mathrm{p}}^3 - 3\kappa_{\mathrm{p}}^2 + 2\kappa_{\mathrm{p}}) & \text{if} \quad \kappa_{\mathrm{p}} < 1 \\ 1 & \text{else} \end{cases}$$

and

$$q_{\mathrm{h2}}(\kappa_{\mathrm{p}}) = \begin{cases} 1 & \text{if} \quad \kappa_{\mathrm{p}} < 1 \\ 1 + H_{\mathrm{p}}(\kappa_{\mathrm{p}} - 1) & \text{else} \end{cases}$$

where q_{h0} defines the initial yield limit, and H_{p} is a dimensionless hardening modulus that controls the hardening in the post-peak regime. Both parameters should be positive and fulfill the inequality $H_{\mathrm{p}} + q_{\mathrm{h0}} < 1$.

DAMAGE MODEL

The definition of the two damage variables, d_{t} and d_{c}, are defined in the framework of isotropic damage mechanics, It can described by a damage loading function $F_{\mathrm{d}i}$ and Kuhn–Tucker loading-unloading conditions

$$F_{\mathrm{d}i} = \varepsilon_{\mathrm{eq}}(\sigma_{\mathrm{un}}) - \kappa_{\mathrm{d}i} \tag{3-124}$$

$$F_{di} \leq 0,\ \dot{\kappa}_{di} \geq 0,\ \dot{\kappa}_{di} F_{di} = 0 \tag{3-125}$$

where the index i = t, c stands for tension or compression, respectively. In Equation 3-124, ε_{eq} stands for the equivalent strain and κ_{di} is a history variable.

The damage variables are then defined as

$$d_i = g_{di}(\kappa_{di}, \kappa_{di1}, \kappa_{di2})$$

where κ_{di1} and κ_{di2} are additional history variables.

The equivalent strain is the same for both tension and compression, and it is derived from the plastic yield surface $F_p = 0$. Using $q_{h1} = 1$ and $q_{h2} = \varepsilon_{eq} / \varepsilon_0$ one can derive

$$\varepsilon_{eq}(\zeta_{un}, r_{un}, \theta_{un}) = \frac{\varepsilon_0 m_0}{2}\left[\frac{r_{un}}{\sqrt{6}\sigma_{cs}} r(\theta_{un}) + \frac{\zeta_{un}}{\sigma_{cs}}\right] + \sqrt{\frac{\varepsilon_0^2 m_0^2}{4}\left[\frac{r_{un}}{\sqrt{6}\sigma_{cs}} r(\theta_{un}) + \frac{\zeta_{un}}{\sigma_{cs}}\right]^2 + \frac{3\varepsilon_0^2 r_{un}^2}{2\sigma_{cs}^2}}$$

where $\varepsilon_0 = \sigma_{ts} / E$. Noteworthy is that the equivalent strain ε_{eq} reduces to

$$\varepsilon_{eq} = \frac{\sigma_{un}}{E}$$

in uniaxial tension, and to

$$\varepsilon_{eq} = \frac{\sigma_{un}}{E}\frac{\sigma_{ct}}{\sigma_{cs}}$$

in uniaxial compression.

History Variables, Tension

The equivalent strain in tension is computed from the evolution equation

$$\dot{\varepsilon}_{eqt} = \dot{\varepsilon}_{eq}$$

or directly as $\varepsilon_{eqt} = \varepsilon_{eq}$. The definition of κ_{dt} then follows from Equation 3-124 and Equation 3-125.

Two additional history variables, κ_{dt1} and κ_{dt2}, are introduced to account for how damage evolves during different stress states. The variable κ_{dt1} is a function of the rate of the plastic strain tensor and it is given by the evolution equation

$$\dot{\kappa}_{dt1} = \begin{cases} \dfrac{1}{\chi_s(\zeta_{un}, r_{un})} \|\dot{\varepsilon}_{pl}\| & \text{if } \dot{\kappa}_{dt} < 1 \text{ and } \kappa_{dt} > \varepsilon_0 \\ 0 & \text{otherwise} \end{cases}$$

The variable κ_{dt2} is a function of κ_{dt} given by

$$\dot{\kappa}_{dt2} = \frac{1}{\chi_s(\zeta_{un}, r_{un})} \dot{\kappa}_{dt}$$

The evolution of these variables, and thus also the softening response of the model, is sensitive to the multiaxial stress state that enters in the equations through the *ductility* measure χ_s. The ductility is defined as

$$\chi_s(\zeta_{un}, r_{un}) = 1 + (A_s - 1)R_s \tag{3-126}$$

with

$$R_s = \begin{cases} -\dfrac{\sqrt{6}\zeta_{un}}{r_{un}} & \text{if } \zeta_{un} < 0 \\ 0 & \text{otherwise} \end{cases}$$

In uniaxial compression, the variables read $R_s = 1$ and $\chi_s = A_s$, which can be useful for calibrating the parameter A_s. COMSOL Multiphysics by default uses $A_s = 10$.

History Variables, Compression

The equivalent strain in compression is given by the evolution equation

$$\dot{\varepsilon}_{eqc} = \alpha_c \dot{\varepsilon}_{eq}$$

The definition of the history variable κ_{dc} then follows after Equation 3-124 and Equation 3-125. The two additional history variables κ_{dc1} and κ_{dc2} are introduced to account for damage evolution at different stress states. The variable κ_{dc1} is a function of the rate of the plastic strain tensor,

$$\dot{\kappa}_{dc1} = \begin{cases} \dfrac{\alpha_c \beta_c}{\chi_s(\zeta_{un}, r_{un})} \|\dot{\varepsilon}_{pl}\| & \text{if } \dot{\kappa}_{dc} < 1 \text{ and } \kappa_{dc} > \varepsilon_0 \\ 0 & \text{otherwise} \end{cases}$$

and the variable κ_{dc2} is a function of κ_{dt} given by

$$\dot{\kappa}_{dc2} = \frac{1}{\chi_s(\zeta_{un}, r_{un})}\dot{\kappa}_{dc}$$

The factor α_c is introduced to distinguish between tensile and compressive stresses, and follows from the spectral decomposition of the undamaged stress σ_{un} as

$$\alpha_c = \sum_{i=1}^{3} \frac{-\langle -\sigma_{un,i}\rangle(\langle \sigma_{un,i}\rangle - \langle -\sigma_{un,i}\rangle)}{\|\sigma_{un}\|^2}$$

where $\sigma_{un,i}$ is the ith principal value of σ_{un}, and the operator $< \cdot >$ denotes the Macaulay brackets. Note that α_c varies from 0 in pure tension to 1 in pure compression.

The factor β_c is introduced to accomplish a smooth transition between the case of pure damage to damage–plasticity softening, which may occur during cyclic loading. It is defined as

$$\beta_c = \sqrt{\frac{2}{3(1+2D_f^2)}}\frac{\sigma_{ct} q_{h2}(\kappa_p)}{r_{un}}$$

where D_f is the same model parameter used for the plasticity flow. The ductility measure χ_s is defined in Equation 3-126.

Damage Variables

Given the history variables, the definition of the damage variables d_t and d_c follows the same derivation. Assuming monotonic and uniaxial loading, the damaged stress in the softening regime is described by an enhanced inelastic strain, $\varepsilon_{inel,d}$,

$$\sigma_{dmg} = (1-d)\sigma_{un} = (1-d)E(\varepsilon - \varepsilon_{inel}) = E(\varepsilon - \varepsilon_{inel,d}) \tag{3-127}$$

The damage variable is then defined such that the post-peak (or softening) is described by an equation of the form

$$\sigma_{dmg} = f_s(\varepsilon_{inel,d}) \tag{3-128}$$

where $f_s(\cdot)$ is the *softening law* often referred to as the *traction–separation law*, here given in a terms of inelastic strains.

Since damage is an irreversible process, it is postulated that the inelastic strain $\varepsilon_{inel,d}$ can be described by the damage history variables. This strain measure is called the *inelastic damage strain* (or crack-opening strain) and it is defined as

$$\varepsilon_{dmg} = \kappa_{d1} + d\kappa_{d2}$$

Replacing $\varepsilon - \varepsilon_{inel}$ in Equation 3-127 with κ_d such that

$$\sigma_{dmg} = (1-d)E\kappa_d$$

and using it together with Equation 3-128 results in an equation for the damage variable in terms of the history variables of the form

$$f_s(\kappa_{d1} + d\kappa_{d2}) = (1-d)E\kappa_d \tag{3-129}$$

This nonlinear equation is solved for d using Newton's method at each material point. COMSOL Multiphysics provides a number of built-in softening functions $f_s(\cdot)$ for both tension and compression:

- Linear
- Bilinear
- Exponential
- Cornelissen (only tension)

These are depicted in a Figure 3-28. The linear and bilinear definitions have a closed-form solution to Equation 3-129, which makes them computationally attractive.

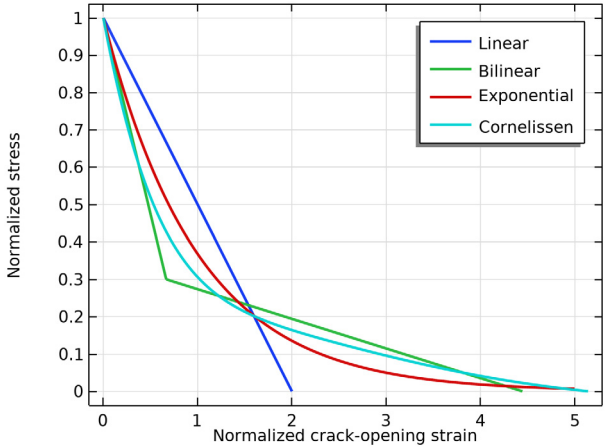


Figure 3-28: Built-in softening laws.

When combined with the crack-band method, the stress versus strain curves in Figure 3-28 are defined in terms of stress versus displacement. The stress versus

displacement curves are then converted to unique stress versus strains curves at each material point using the crack-band width h_{cb}, see The Crack Band Method.

In addition to the built-in softening laws, COMSOL Multiphysics allows user-defined definitions of $f_s(\cdot)$. Use the variables `<phys>.<feat>.<feat>.edmgt` and `<phys>.<feat>.<feat>.edmgc` to defined the damaged strain in tension and compression, respectively. The user-defined function should resemble that in Figure 3-28 with stress and strain interpreted as positive variables in tension and compression.

Alternatively, the user-defined softening behavior can be specified in terms of a stress versus displacement curve, $f_w(\cdot)$. Equation 3-129 is then replaced by

$$f_w(w_{dmg}) = f_w[(\kappa_{d1} + d\kappa_{d2})h_{cb}] = (1-d)E\kappa_d$$

assuming that the crack-band method is used. COMSOL Multiphysics provides the variables `<phys>.<feat>.<feat>.wdmgt` and `<phys>.<feat>.<feat>.wdmgc` for the damaged displacement in tension and compression, respectively. For both options, the nonlinear equation is solved to find d.

Elastoplastic Soils

In this section:

- The Modified Cam-Clay Soil Model
- The Modified Structured Cam-Clay Soil Model
- The Extended Barcelona Basic Soil Model
- The Hardening Soil Model
- The Hardening Soil Small Strain Model

THE MODIFIED CAM-CLAY SOIL MODEL

The Cam-clay material model was developed in the 1970s at the University of Cambridge, and since then it has experienced many modifications. The *modified Cam-clay* (MCC) model is the most commonly used model due to its smooth yield surface, and it is the one implemented in COMSOL Multiphysics.

The modified Cam-clay model is a so-called *critical state* model, where the loading and unloading of the material follow different paths in stress space. The model also features hardening and softening. Different formulations can be found in textbooks, see for instance Ref. 68, Ref. 69, and Ref. 70.

The yield function was originally written in terms of the Invariants of the Stress Tensor

$$p = -\frac{I_1(\sigma)}{3}$$
$$q = \sqrt{3J_2(\sigma)}$$
$$\theta = \frac{1}{3}\text{acos}\left(\frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{3/2}}\right)$$

Following the Structural Mechanics Module sign convention (the pressure is positive in compression), the yield function reads:

$$F_y = q^2 + M^2(p - p_c)p = 0$$

This is an ellipse in the Meridional Plane, with a circular section in the Octahedral Plane.

If we write the Cam-clay yield function in terms of the invariants described in Pressure-Dependent and Soil Plasticity:

$$\sigma_m = \frac{I_1}{3} = -p$$
$$\sigma_{mises} = \sqrt{3J_2} = q$$

we get

$$F_y = \Gamma(\theta)^2\sigma_{mises}^2 + M^2(\sigma_m + p_c)\sigma_m = 0$$

Here, σ_m is the *mean stress*, σ_{mises} is the von Mises *equivalent stress*, θ is the *Lode angle*, and $\Gamma(\theta)$ is a function of θ that defines the shape of the yield function in the Octahedral Plane.

The shape of the yield function in the Octahedral Plane, $\Gamma(\theta)$, can be made a function of the Lode angle θ by using any of the functions described in Octahedral Section. The most common choice, $\Gamma(\theta) = 1$, results in a circular section that recovers the original Cam-clay model.

The parameter $p_c > 0$ is the *consolidation pressure*, and the parameter $M > 0$ defines the slope of the *critical state line* in the Meridional Plane which also represents the eccentricity of the ellipse. This parameter can be matched to the friction angle ϕ in the Mohr–Coulomb Criterion criterion as

$$M = \frac{6\sin(\phi)}{3 - \sin(\phi)} \tag{3-130}$$

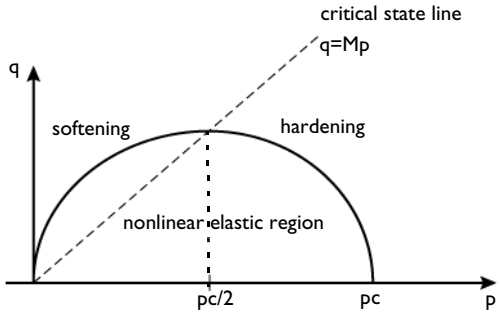


Figure 3-29: Modified Cam-clay ellipse in the meridional plane. The ellipse circumscribes a nonlinear elastic region.

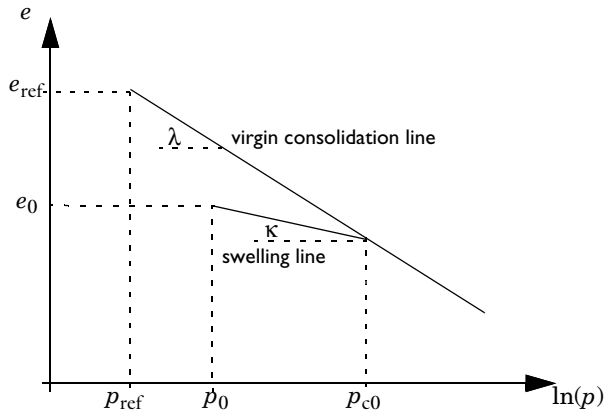
The Cam-clay criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the plastic potential is defined as

$$Q_{\mathrm{p}} = \Gamma_Q^2 \sigma_{\mathrm{mises}}^2 + M_Q^2 (\sigma_{\mathrm{m}} + p_{\mathrm{c}}) \sigma_{\mathrm{m}}$$

The parameter M_Q represents the slope of the *critical state line* for the plastic potential, which a can be matched to the Mohr–Coulomb Criterion criterion at the compressive meridian by the dilatation angle ψ

$$M_Q = \frac{6\sin(\psi)}{3 - \sin(\psi)}$$

The soil response to isotropic compression is described by the curve of the void ratio (or specific volume or volumetric strain) versus the logarithm of pressure as shown in Figure 3-30.



The void ratio e is the ratio between the pore space and solid volume. It can be written in terms of the porosity ϕ as $e = \phi/(1-\phi)$.

Figure 3-30: Slopes of the virgin isotropic consolidation line and swelling line in the e versus $\ln(p)$ *plane.*

The compression index λ corresponds to the slope of the *virgin isotropic consolidation line*, and the swelling index κ corresponds to the slope of the *swelling line* (also called unloading-reloading line) in the e versus $\ln(p)$ plane. The reference void ratio e_{ref} is measured at the reference pressure p_{ref} and the initial void ratio e_0 is measured at the initial mean stress p_0. The starting value of the initial mean stress p_0 is the reference pressure p_{ref}. The equation for the virgin isotropic consolidation line reads

$$e = e_{\text{ref}} - \lambda \ln\left(\frac{p}{p_{\text{ref}}}\right)$$

The void ratio at the preconsolidation pressure p_{c0} is given by

$$e_{c0} = e_{\text{ref}} - \lambda \ln\left(\frac{p_{c0}}{p_{\text{ref}}}\right)$$

The equation for the swelling line is

$$e = e_0 - \kappa \ln\left(\frac{p}{p_0}\right)$$

where e_0 is the initial void ratio calculated as

$$e_0 = e_{c0} + \kappa \ln\left(\frac{p_{c0}}{p_0}\right)$$

The initial void ratio e_0 can be directly entered as user input. In this case, the void ratio at the initial consolidation pressure, e_{c0}, is not needed as intermediate variable.

Assuming small deformations and adopting the sign convention of Structural Mechanics, the volumetric strain is related to the void ratio as

$$\varepsilon_{vol} = \frac{V - V_0}{V_0} = \frac{e - e_0}{1 + e_0}$$

For elastic response, the incremental volumetric elastic strain is written as

$$d\varepsilon_{el,vol} = \frac{de}{1 + e_0} = -\frac{\kappa}{1 + e_0}\frac{dp}{p} \tag{3-131}$$

Integrating Equation 3-131, the equation is reformulated as

$$\varepsilon_{el,vol} = -\frac{\kappa}{1 + e_0} \ln\left(\frac{p}{p_0}\right)$$

Then the nonlinear relation between pressure and volumetric strain is obtained by

$$p = p_0 e^{-B_{el}\varepsilon_{el,vol}} \tag{3-132}$$

where

$$B_{\mathrm{el}} = \frac{1+e_0}{\kappa}$$

The contribution to initial mean stress p_0 also comes from the mean stress in the initial or external stress tensor. The reference pressure is generally a unit pressure in the used unit system, in the literature the value varies from 1 kPa to 100 kPa. In COMSOL Multiphysics the default value is 100 kPa.

At zero volumetric elastic strain and with zero contribution from the Initial Stress and Strain or External Stress features, the initial mean stress is nonzero and equal to the reference pressure p_{ref}. The reference pressure acts as an *in situ* stress.

$$p_0 = p_{\mathrm{ref}} \tag{3-133}$$

The tangent bulk modulus K_{t} is written as

$$K_{\mathrm{t}} = \frac{1+e_0}{\kappa} p$$

The secant bulk modulus K_{s} (or bulk modulus K) is written as

$$K_s = \frac{p_0 - p}{\varepsilon_{\mathrm{el,vol}}}$$

When a constant Poisson's ratio is specified, the shear modulus is calculated from the Poisson's ratio and secant bulk modulus, which in turn gives a variable shear modulus.

The total volumetric strain increment in written as

$$d\varepsilon_{\mathrm{vol}} = \frac{de}{1+e_0} = -\frac{\kappa}{1+e_0}\frac{dp}{p} - \frac{\lambda-\kappa}{1+e_0}\frac{dp_{\mathrm{c}}}{p_{\mathrm{c}}}$$

The evolution of the void ratio is then written as

$$e = (1+e_0)\varepsilon_{\mathrm{vol}} + e_0 \text{ or} \tag{3-134}$$

$$e = e_{\mathrm{ref}} - \kappa \ln\left(\frac{p}{p_0}\right) - \lambda \ln\left(\frac{p_{\mathrm{c}}}{p_{\mathrm{ref}}}\right) + \kappa \ln\left(\frac{p_{\mathrm{c}}}{p_0}\right) \tag{3-135}$$

Equation 3-134 is used to compute the void ratio, which matches the analytical expression in Equation 3-135. These two equations can be used for verification purposes.

In the modified Cam-clay model, hardening is controlled by the *consolidation pressure* p_c, which depends on the volumetric plastic strain $\varepsilon_{p,vol}$ as

$$dp_c = -\frac{1+e_0}{\lambda-\kappa} p_c d\varepsilon_{p,vol} \tag{3-136}$$

The elastic and plastic volumetric strains are available in the variable `solid.eelvol` and `solid.epvol`, respectively. The consolidation pressure is available in the variable `item.pc`.

The evolution of the consolidation pressure depends on the values for the *initial void ratio* e_0, the *swelling index* κ, and the *compression index* λ, which are positive parameters that fulfill

$$0 < \kappa < \lambda \text{ and } e_{ref} > 0$$

If an Initial Stress and Strain or External Stress node is added, the initial consolidation pressure p_{c0} must be larger than the mean initial (or external) stress, otherwise the initial stress state will be outside the Cam-clay yield surface.

Volumetric Elastic Deformation

The stress–strain relation beyond the elastic range is of great importance in soil mechanics. For additive decomposition of strains, Cauchy stress tensor is written as

$$\sigma = p_0 e^{-B_{el}\varepsilon_{el,vol}} + 2G\,\mathrm{dev}(\varepsilon_{el})$$

with

$$B_{el} = \frac{1+e_0}{\kappa}$$

Here, σ is the stress tensor, ε is the total strain tensor, ε_{el} is the elastic strain tensor, and G is the shear modulus.

At zero volumetric strain, and with zero contribution from Initial Stress and Strain or External Stress features, the pressure in the MCC model is equal to the reference pressure p_{ref} (see Equation 3-132 and Equation 3-133). This is needed since the MCC model does not have stiffness at zero stress. The reference pressure appears as an additional term in the variational formulation (weak equation).

The modified Cam-clay model introduces a nonlinear relation for the hydrostatic pressure as a function of the volumetric elastic strain given by Equation 3-132.

Hardening and Softening

The yield surface for the original Cam-clay model reads

$$F_{\text{y}} = q^2 + M^2(p - p_{\text{c}})p$$

The plastic strain tensor ε_{p} is computed from the flow rule

$$\dot{\varepsilon}_{\text{p}} = \lambda_{\text{p}} \frac{\partial Q_{\text{p}}}{\partial \sigma}$$

where λ_{p} is the plastic multiplier and the derivatives of the plastic potential Q_{p} with respect to the stress tensor σ. An associated flow rule is used such that $Q_{\text{p}} = F_{\text{y}}$.

The plastic flow rule simplifies to

$$\dot{\varepsilon}_{\text{p}} = \lambda_{\text{p}} \left(3\,\text{dev}(\sigma) - \frac{1}{3} M^2 (2p - p_{\text{c}}) I \right)$$

where the plastic strain rate tensor $\dot{\varepsilon}_{\text{p}}$ includes both deviatoric and volumetric parts. The trace of the plastic strain rate tensor (the *volumetric plastic strain rate* $\dot{\varepsilon}_{\text{pvol}}$) then reads

$$\dot{\varepsilon}_{\text{pvol}} = \text{trace}(\dot{\varepsilon}_{\text{p}}) = \lambda_{\text{p}} M^2 (p_{\text{c}} - 2p)$$

This relation explains why there is *isotropic hardening* when the pressure is $p > p_{\text{c}}/2$ and *isotropic softening* when $p < p_{\text{c}}/2$. As opposed to what happens in Metal Plasticity, in the modified Cam-clay soil model the volumetric plastic strain can either increase or decrease as plastic deformation occurs.

In the MCC model, hardening is controlled by changes in the consolidation pressure p_{c} as a function of volumetric plastic strain.

Hardening introduces changes in the shape of the Cam-clay ellipse, since its major semiaxis depends on the value of the consolidation pressure p_{c}. The *initial*

consolidation pressure p_{c0} defines the size of the ellipse before plastic deformation occurs.

See also Additive Plastic Flow and Numerical Integration Algorithm. If a multiplicative decomposition of strains is used, the flow rule is reformulated according to Multiplicative Plastic Flow.

Including Pore Pressure

When an external pore pressure p_f is added to the MCC material, the yield function is shifted along the p axis.

The quantity $p - p_f$ is normally regarded as the *effective pressure*, or effective stress, which should not be confused with the equivalent von Mises stress. To add the effect of a fluid pressure in the pores to the modified Cam-clay material, add an External Stress feature.

See also the description of the Elastoplastic Soil Material materials in the Solid Mechanics interface documentation.

THE MODIFIED STRUCTURED CAM-CLAY SOIL MODEL

The *structured Cam-clay* (SCC) model (Ref. 76) was developed to circumvent the limitations of the Cam-clay model when applied to structured soils and clays. The SCC model, however, does not consider the influence of the soil structure neither on strength characteristics (especially cohesion) nor in the softening behavior, and it is also not suitable to model cemented clays.

The *modified structured Cam-clay* (MSCC) model was further developed to model destructured, naturally structured and artificially structured clays (Ref. 75), and it is the one implemented in COMSOL Multiphysics.

In the MSCC model, the reduction of mean effective stress due to structure degradation, p_b, depends on the equivalent plastic strain ε_{pe}. The destructuring mechanism is the process of reducing structure strength due to the degradation and crushing of the structure. The structure degradation is given by

$$p_b = \begin{cases} p_{b0} e^{-\varepsilon_{pe}} & \text{if } \varepsilon_{pe} \le \varepsilon^f_{p,dev} \\ p_{bf} e^{-d_s(\varepsilon_{pe} - \varepsilon^f_{p,dev})} & \text{otherwise} \end{cases} \tag{3-137}$$

Here, p_{b0} is the initial structural strength, p_{bf} the failure structural strength, $\varepsilon^{f}_{p,dev}$ is the plastic deviatoric strain at failure, and d_s is the *destructuring index* due to shear deformation.

The equivalent deviatoric plastic strain at which the crushing of the structure begins, $\varepsilon_{p,devc}$, has a typical value between 0.15 and 0.3 for most clays.

Structured clays show a higher void ratio than destructured clays when loaded at the same effective stress. The virgin compression behavior during the destructuring process is expressed by

$$e = e^* + \Delta e$$

$$\Delta e = \Delta e_{c0}\left(\frac{p_{c0}}{p_c}\right)^{d_v}$$

where is e is the void ratio of the structured clay, Δe is the *additional void ratio*, and e^* is the *void ratio of the destructured clay* at the same stress state. The void ratio is computed by using Equation 3-135. Further, Δe_{c0} is the *additional void ratio at preconsolidation pressure*, and d_v is the *destructuring index* due to volumetric deformation.

The yield surface for the modified structured Cam-clay model reads

$$F_y = \Gamma(\theta)^2 q^2 + M^2 (p + p_b)(p - p_c)$$

Here, θ is the *Lode angle*, and $\Gamma(\theta)$ is a function of θ that defines the shape of the yield function in the Octahedral Plane.

The shape of the yield function in the Octahedral Plane, $\Gamma(\theta)$, can be made a function of the Lode angle θ by using any of the functions described in Octahedral Section. The most common choice, $\Gamma(\theta) = 1$, results in a circular section that recovers the original criterion.

The modified structured Cam-clay criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the McDowell–Hau plastic potential is defined as

$$Q_p = \Gamma_Q^2 q^2 + \frac{M_Q^2}{\varsigma - 1}\left((p + p_b)^2 - \left(\frac{p + p_b}{p_c + p_b}\right)^{2/\varsigma}(p_c + p_b)^2\right)$$

The parameter M_Q represents the slope of the *critical state line* for the plastic potential, which a can be matched to the Mohr–Coulomb Criterion criterion at the compressive meridian by the dilatation angle ψ

$$M_Q = \frac{6\sin(\psi)}{3-\sin(\psi)}$$

Here, ς is a parameter to smooth the shape of the plastic potential. The slope of the critical state line, M_Q, is found by using Equation 3-130.

In the modified structured Cam-clay soil model, hardening is controlled by the consolidation pressure p_c as a function of volumetric plastic strain, and it is described as

$$\dot{p}_c = \begin{cases} -\dfrac{1+e_0}{\lambda^* - \kappa + d_v \Delta e\left(\dfrac{M}{M-\eta}\right)} p_c \dot{\varepsilon}_{p,vol} & \text{if } \eta < M \\ -\dfrac{1+e_0}{\lambda^* - \kappa + d_v \Delta e} p_c \dot{\varepsilon}_{p,vol} & \text{otherwise} \end{cases}$$

Here, λ^* is the *compression index for destructured clay*, κ is the *swelling index*, and η is ratio of shear stress to mean stress.

The MSCC and MCC models are equivalent when $\Delta e_{c0} = 0$, $\varsigma = 2$, and $p_{b0} = 0$.

The relationship between hydrostatic pressure and volumetric elastic strain is outlined in The Modified Cam-Clay Soil Model,

$$p = p_0 e^{-B_{el}\varepsilon_{el,vol}} \text{ with } B_{el} = \frac{1+e_0}{\kappa} \tag{3-138}$$

The stress tensor is then computed from

$$\sigma = p_0 e^{-B_{el}\varepsilon_{el,vol}} + 2G\,\mathrm{dev}(\varepsilon_{el})$$

At zero volumetric strain, and with zero contribution from the Initial Stress and Strain or External Stress features, the pressure in the MSCC model is equal to the reference pressure p_{ref} (see Equation 3-138 and Equation 3-133). This is needed since the MSCC model does not have stiffness at zero stress. The reference pressure appears as an additional term in the variational formulation (weak equation).

THE EXTENDED BARCELONA BASIC SOIL MODEL

The *Barcelona basic model* (BBM) was developed to simulate the loading of unsaturated and partially saturated soils, by incorporating an extra state variable for the pore suction. The suction value depends on the amount of water in the soil, and it affects the flow in porous soils as well as the deformation and stress distribution.

The BBM model uses the concepts of plasticity theory, incorporating the critical state model (Ref. 77). This soil model matches the results obtained with the modified Cam-clay model in fully saturated soils.

The so-called *extended Barcelona basic model* (BBMx) was further developed to overcome numerical limitations in the original BBM model (Ref. 78). The model implemented in COMSOL Multiphysics follows Ref. 78 with certain modifications described in this section.

The BBMx model presents a smooth yield surface with respect to both stress and suction

$$F_y = \Gamma(\theta)^2 q^2 + M^2 (p - p_{cs})(p + p_s) + p_{ref}^2 \left(e^{\frac{b(s - s_y)}{p_{ref}}} - e^{\frac{-bs_y}{p_{ref}}} \right)$$

Here, p and q are stress invariants as defined in The Modified Cam-Clay Soil Model, p_{cs} is the consolidation pressure at current suction, p_s is the tensile strength due to current suction, s is the current suction, b is a dimensionless smoothing parameter, s_y is the yield value at current suction, and p_{ref} is the reference pressure at which the reference void ratio e_{ref} was measured.

The shape of the yield function in the Octahedral Plane, $\Gamma(\theta)$, can be made a function of the Lode angle θ by using any of the functions described in Octahedral Section. The most common choice, $\Gamma(\theta) = 1$, results in a circular section that recovers the original criterion.

The tensile strength due to current suction, p_s, is linearly related to the suction level as $p_s = ks$, where k is the tension to suction ratio.

The BBMx criterion can be used with either an associated or nonassociated flow rule. For the nonassociated flow rule, the plastic potential is defined as

$$Q_p = \Gamma_Q^2 q^2 + M_Q^2 (p - p_{cs})(p + p_s) + p_{ref}^2 \left(e^{\frac{b(s - s_y)}{p_{ref}}} - e^{\frac{-bs_y}{p_{ref}}} \right)$$

The parameter M_Q represents the slope of the *critical state line* for the plastic potential, which a can be matched to the Mohr–Coulomb Criterion criterion at the compressive meridian by using the dilatation angle ψ

$$M_Q = \frac{6\sin(\psi)}{3-\sin(\psi)}$$

Here, ς is a parameter to smooth the shape of the plastic potential. The slope of the critical state line, M_Q, is found by using Equation 3-130.

The consolidation pressure at current suction p_{cs} is calculated from

$$p_{cs} = p_{ref}\left(\frac{p_c}{p_{ref}}\right)^{\left(\frac{\lambda_0-\kappa}{\lambda(s)-\kappa}\right)}$$

where $\lambda(s)$ is the compression index at current suction, λ_0 is the compression index at saturation, and κ is the swelling index. The compression index at current suction, $\lambda(s)$, is given by

$$\lambda(s) = \lambda_0((1-w)e^{-s/m}+w)$$

where w and m are weighting and soil stiffness parameters.

The slope of the critical state line M can be computed from the Matsuoka–Nakai criterion, in which case it depends on both the Lode angle θ and the angle of internal friction ϕ

$$M = \left(\frac{6\sin(\phi)}{3-\sin(\phi)}\right)\left(\frac{2\omega}{1+\omega-(1-\omega)\sin\left(3\theta-\frac{\pi}{2}\right)}\right)^{1/4} \tag{3-139}$$

where

$$\omega = \left(\frac{3-\sin(\phi)}{3+\sin(\phi)}\right)^4$$

At the tensile or compressive meridians, where the Lode angle is $\theta = 0$ or $\theta = \pi/3$, the slope of the critical state line achieves the same expression as when matched to the Mohr–Coulomb Criterion, see Equation 3-130

$$M = \frac{6\sin(\phi)}{3-\sin(\phi)}$$

It can also be matched to the Matsuoka–Nakai Criterion.

The associated plastic potential for the BBMx model reads

$$Q_{\mathrm{p}} = F_{\mathrm{y}}$$

The plastic strain increments are computed from the derivatives of the plastic potential with respect to stress only.

As in The Modified Cam-Clay Soil Model, hardening is controlled by the evolution of the consolidation pressure p_{c}, which depends on the volumetric plastic strain $\varepsilon_{\mathrm{p,vol}}$.

$$dp_{\mathrm{c}} = -\frac{1+e_0}{\lambda_0-\kappa}p_{\mathrm{c}}d\varepsilon_{\mathrm{p,vol}} \tag{3-140}$$

The initial void ratio e_0, the swelling index κ, and the compression index at saturation λ_0, are positive parameters.

The evolution of the yield value at current suction, s_{y}, is also governed by the volumetric plastic strain $\varepsilon_{\mathrm{pl,vol}}$ as

$$ds_{\mathrm{y}} = -\frac{1+e_0}{\lambda_s-\kappa_s}(s_{\mathrm{y}}+p_{\mathrm{atm}})d\varepsilon_{\mathrm{p,vol}} \tag{3-141}$$

Here, λ_{s} is the compression index for changes in suction, κ_{s} is the swelling index for changes in suction, and p_{atm} is the atmospheric pressure, all positive parameters. Note that Equation 3-140 and Equation 3-141 are normally given with opposite sign. Here, however, the structural mechanics convention is used, so the increments in consolidation pressure and suction are positive in compression.

The evolution of void ratio is then written as

$$e = (1+e_0)\varepsilon_{\mathrm{vol}} + e_0 \text{ or}$$

$$e = e_{s,\,\mathrm{ref}} - \lambda(s)\ln\left(\frac{p_{cs}}{p_{\mathrm{ref}}}\right) + \kappa\ln\left(\frac{p_{cs}}{p_{\mathrm{i}}}\right) - \kappa\ln\left(\frac{p}{p_{\mathrm{i}}}\right)$$

where

$$e_{s,\,\mathrm{ref}} = e_{\mathrm{ref}} - \kappa_s\ln((s+p_{\mathrm{atm}})/p_{\mathrm{atm}})$$

In the BBMx model, the total volumetric elastic response is combination of elastic response by pressure and suction,

$$d\varepsilon_{\text{el,vol}} = d\varepsilon^{\sigma}{}_{\text{el,vol}} + d\varepsilon^{s}{}_{\text{el,vol}} = -\frac{dp}{K} - \frac{ds}{K_{\text{cs}}}$$

where

$$K_{\text{cs}} = (1 + e_0)(s + p_{\text{atm}})/\kappa_s$$

Here K is the bulk modulus, and K_{cs} the stiffness to suction. Note that the pressure p in compression and suction s is positive variables, but the elastic volumetric strain $\varepsilon_{\text{el,vol}}$ is negative in compression. The volumetric elastic response due to suction is given by

$$\varepsilon^{s}{}_{\text{el,vol}} = -\frac{\kappa_s}{1 + e_0} \ln\left(\frac{s + p_{\text{atm}}}{s_0 + p_{\text{atm}}}\right)$$

Here, s_0 is the initial suction. The volumetric elastic response due to suction is accounted as an inelastic strains in the total strain tensor, any changes in suction while keeping a constant pressure can cause volumetric strains.

The relationship between hydrostatic pressure and volumetric elastic strain is the same as outlined for The Modified Cam-Clay Soil Model

$$p = p_0 e^{-B_{\text{el}}\varepsilon_{\text{el,vol}}} \text{ with } B_{\text{el}} = \frac{1 + e_0}{\kappa} \tag{3-142}$$

The stress tensor is then computed from

$$\sigma = p_0 e^{-B_{\text{el}}\varepsilon_{\text{el,vol}}} + 2G\,\text{dev}(\varepsilon_{\text{el}})$$

At zero volumetric strain, and with zero contribution from the Initial Stress and Strain or External Stress feature, the pressure in the BBMx model is equal to the reference pressure (see Equation 3-142 and Equation 3-133). This is needed since the BBMx

model does not have stiffness at zero stress. The reference pressure appears as an additional term in the variational formulation (weak equation form).

The material property λ_0 is the compression index at saturation, which does not depend on suction.

The variable $\lambda(s)$, which is a function of the current suction, corresponds to the compression index (slope) in the void ratio versus mean stress plot (log axis).

The material property λ_s corresponds to the compression index (slope) in the void ratio versus matrix suction plot (log axis), which does not depend on the mean stress.

THE HARDENING SOIL MODEL

The *hardening soil* model is an elastoplastic material model with a stress and stress path-dependent stiffness. It is a so-called *double stiffness model*, meaning that the soil stiffness is different during primary loading and unloading/reloading load paths (Ref. 79).

The yield surface for the hardening soil model is a combination of a conical surface and an elliptic cap surface in stress space, and it has separate mechanisms for controlling the soil's shear and volumetric behavior.

In the original hardening soil model, failure in shear occurs according to the Mohr–Coulomb Criterion. In the so-called *hardening soil smooth* model (Ref. 79), failure in shear occurs according to Matsuoka–Nakai Criterion.

In Ref. 81, the shear hardening and failure surfaces are combined, and the failure surface imposes the final limit on shear stresses. COMSOL Multiphysics implements the latter version.

The 50% *failure stiffness* for primary loading, E_{50}, and the *stiffness for unloading and reloading*, E_{ur}, are used to define the elastic relation and hardening variable. These are given by

$$E_{50} = E_{50}^{\text{ref}}\left(\frac{c\cot(\phi) - \sigma_1}{c\cot(\phi) + p_{\text{ref}}}\right)^m$$

and

$$E_{ur} = E_{ur}^{ref}\left(\frac{c\cot(\phi) - \sigma_1}{c\cot(\phi) + p_{ref}}\right)^m$$

Here E_{50}^{ref} is the *reference failure stiffness* for primary loading at reference pressure, E_{ur}^{ref} is the *reference stiffness for unloading and reloading* at reference pressure, c is the cohesion, ϕ is the friction angle, and m is the stress exponent.

The ultimate deviatoric stress q_f and the stress to failure q_a are derived from the Mohr–Coulomb Criterion

$$q_f = \frac{2c\cos(\phi) - 2\sigma_1\sin(\phi)}{1 - \sin(\phi)} \text{ and } q_a = \frac{q_f}{R_f}$$

where R_f is the failure ratio.

Consider the stress invariants $\sigma_{mises} = \sqrt{3J_2}$ and $\sigma_m = I_1/3$, the yield function and plastic potential for shear hardening are given by

$$F_y = \Gamma(\theta)\sigma_{mises} + \frac{M(\phi)}{R_f}(\sigma_m - c\cot(\phi))$$

$$Q_p = \Gamma_Q(\theta)\sigma_{mises} + \frac{M_Q(\psi_m)}{R_f}(\sigma_m - c\cot(\phi))$$

Here, θ is the *Lode angle*, and $\Gamma(\theta)$ is a function of θ that defines the shape of the yield function in the Octahedral Plane, and ψ_m is the *mobilized dilatancy angle*.

The shape of the yield function in the Octahedral Plane, $\Gamma(\theta)$, can be made a function of the Lode angle θ by using any of the functions described in Octahedral Section. The most common choice, $\Gamma(\theta) = 1$, results in a circular section.

The yield function for the elliptic cap is described in the Compression Cap section.

For linear hardening, the internal variable for the preconsolidation pressure, p_c, depends on the volumetric plastic strain $\varepsilon_{pl,vol}$ as

$$\dot{p}_c = -K_c h \dot{\varepsilon}_{pl,vol}$$

where K_c is the hardening modulus and

$$h = \frac{E_{ur}^{ref}}{E_{ur}} = \left(\frac{c\cot(\phi) - \sigma_1}{c\cot(\phi) + p_{ref}}\right)^m$$

The dilatancy cutoff is implemented by setting the *mobilized dilatancy angle* ψ_m equal to zero when the void ratio reaches the critical void ratio e_{max}.

Cauchy's stress tensor is then written as

$$\sigma - \sigma_0 = \mathbf{C}:(\varepsilon - \varepsilon_{inel})$$

where $\mathbf{C}$ is a function of the stiffness modulus E_{ur} and Poisson's ratio ν.

If required, add a Compression Cap or a Tension Cutoff criterion to the *hardening soil* model.

Mobilized Dilatancy Angle

The *mobilized dilatancy angle* ψ_m is explicitly used in the definition for the plastic potential Q_p. Different authors define this variable using different expressions. The original expression for the mobilized dilatancy derived by Rowe (Ref. 79) is

$$\sin\psi_m = \frac{\sin\phi_m - \sin\phi_c}{1 - \sin\phi_m \sin\phi_c} \tag{3-143}$$

here, ϕ_c is the *critical friction angle* and ϕ_m is the *mobilized friction angle*, and these are derived from

$$\sin\phi_c = \frac{\sin\phi - \sin\psi}{1 - \sin\phi\sin\psi}$$

$$\sin\phi_m = \frac{\sigma_1 - \sigma_3}{\sigma_1 + \sigma_3 + 2c\cot\phi}$$

where ψ is the dilatancy angle, σ_i are the principal stresses, c is the cohesion, and ϕ is the angle of internal friction.

Rowe's original model for the mobilized dilatancy angle is highly contractive at low angles, so the modified Rowe's model reads

$$\sin\psi_m = \max\left(0, \frac{\sin\phi_m - \sin\phi_c}{1 - \sin\phi_m \sin\phi_c}\right) \tag{3-144}$$

which however, does not fully account for the plastic volumetric contraction either.

Other expressions (Ref. 79) for the mobilized dilatancy angle include the Wehnert correction

$$\sin\psi_m = \max\left(0{,}07, \frac{\sin\phi_m - \sin\phi_c}{1 - \sin\phi_m \sin\phi_c}\right) \tag{3-145}$$

the Soreide correction

$$\sin\psi_m = \left(\frac{\sin\phi_m - \sin\phi_c}{1 - \sin\phi_m \sin\phi_c}\right)\left(\frac{\sin\phi_m}{\sin\phi}\right) \tag{3-146}$$

THE HARDENING SOIL SMALL STRAIN MODEL

The *hardening soil small strain* model is a combination of Small-Strain Overlay model and The Hardening Soil Model.

The small strain shear modulus is computed from other material parameters, as done in the hardening soil model

$$G_0 = G_0^{\text{ref}}\left(\frac{c\cot(\phi) - \sigma_1}{c\cot(\phi) + p_{\text{ref}}}\right)^m$$

Here G_0^{ref} is the reference small strain shear modulus for primary loading at reference pressure, c is the cohesion, ϕ is the friction angle, and m is the stress exponent.

The *critical shear strain*, defined as

$$\gamma_c = \gamma_{\text{ref}}\left(\frac{G_0^{\text{ref}}}{G_{\text{ur}}^{\text{ref}}} - 1\right)$$

is used to switch between the shear stiffness derived from the small-strain overlay and the hardening soil models:

$$G = \begin{matrix} G_0 & \text{when } \gamma < \gamma_c \\ G_{\text{ur}} & \text{when } \gamma \geq \gamma_c \end{matrix}$$

Here, G_{ur} is the shear modulus for unloading/reloading, as defined in The Hardening Soil Model.

$$G_{\text{ur}} = G_{\text{ur}}^{\text{ref}}\left(\frac{c\cot(\phi) - \sigma_1}{c\cot(\phi) + p_{\text{ref}}}\right)^m$$

The yield function for the elliptic cap is described in the Compression Cap section. If required, add a Tension Cutoff criterion to the *hardening soil small strain* model.

Creep and Viscoplasticity

In this section:

- Creep
  - Creep Material Models
  - Creep Rate Function
  - Equivalent Stress, Creep Flow and Hardening Rule
  - Hardening Function
  - Thermal Function
  - Combining Creep Models
  - Time Integration
  - Converting Between Different Data Representations
- Viscoplasticity
  - Anand Model
  - Chaboche Model
  - Perzyna Model
  - Bingham Model
  - Peric Model
- Creep and Viscoplasticity for Large Strains
- Polymer Viscoplasticity
  - Bergstrom–Boyce Model
  - Bergstrom–Bischoff Model
  - Parallel Network Model
- Energy Dissipation

In the literature, the terms *viscoplasticity* and *creep* are often used interchangeably to refer to the class of problems related to rate-dependent plasticity. A distinction is sometimes made so that *creep* refers to models without a yield surface.

CREEP

Creep is an inelastic time-dependent deformation that occurs when a material is subjected to stress (typically much less than the plastic yield stress) at sufficiently high temperatures.

The creep strain rate depends in a general case on stress, temperature, and time, usually in a nonlinear manner:

$$\dot{\varepsilon}_{cr} = F_{cr}(\sigma, T, t)$$

It is often possible to separate these effects multiplicatively:

$$\dot{\varepsilon}_{cr}(\sigma, t, T) = f(\sigma)g(T)h(t)$$

Experimental data shows three different types of behavior for the creep strain rate at constant stress as function of time. Researchers normally subdivide the creep curve into these three regimes, based on the fact that many different materials show similar responses:

- In the initial *primary creep* regime (also called *transient creep*) the creep strain rate decreases with time to a minimum steady-state value.
- In the *secondary creep* regime the creep strain rate is almost constant. This is also called *steady-state* creep.
- In the *tertiary creep* regime the creep strain increases with time until a failure occurs.

When this distinction is assumed, the total creep rate can be additively split into primary, secondary, and tertiary creep rates

$$\dot{\varepsilon}_{cr} = F_{cr1} + F_{cr2} + F_{cr3}$$

In most cases, F_{cr1} and F_{cr3} depend on stress, temperature, and time, while secondary creep, F_{cr2}, depends only on stress level and temperature. Normally, secondary creep

is the dominant process. Tertiary creep is seldom important because it only accounts for a small fraction of the total lifetime of a structure or mechanical component.

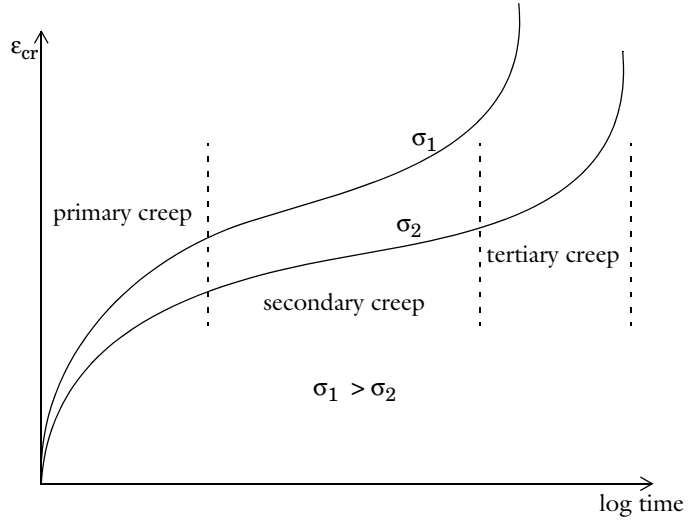


Figure 3-31: Uniaxial creep as a function of logarithmic time.

Creep Material Models

The mathematical description of a general creep model can be summarized by a set of equations given in a similar format as the flow theory of plasticity. The creep strain is given by a flow rule that defines the relationship between the rate of the creep strain tensor ε_{cr} and the current state of stress. This flow rule can be written as

$$\dot{\varepsilon}_{cr} = \lambda_{cr} \frac{\partial Q_{cr}}{\partial \sigma}$$

where λ_{cr} is the creep multiplier, Q_{cr} is the creep potential, and σ is the stress tensor. The creep multiplier is an explicit function of stress σ, temperature T, time t, and the equivalent creep strain ε_{ce}, and can be given on a general format as

$$\lambda_{cr} = f(\sigma_e) g(T) h(\varepsilon_{ce}, t)$$

where the generic functions f, g, and h that define the creep rate and can be combined to construct different types of creep models. The hardening rule, $h(\varepsilon_{ce}, t)$, defines the relationship between the creep strain tensor ε_{cr} and the equivalent creep strain ε_{ce}. The equivalent stress σ_e is a scalar representation of the Cauchy stress tensor, and given associative creep flow, it also defines the creep potential so that

$$Q_{cr} = \sigma_e$$

Creep Rate Function

The creep rate function f is typically a function of the equivalent stress σ_e or any other scalar measure of the current stress state. In COMSOL Multiphysics, there are several built-in creep rate functions:

- Norton (Power law)
- Garofalo (Hyperbolic sine law)
- Nabarro–Herring (Diffusional Creep)
- Coble (Diffusional Creep)
- Weertman (Dislocation Creep)

It is also possible to define an arbitrary expression for f so that, in principle, any creep model can be implemented. The expressions used for the built-in creep rate functions are described in the following sections.

The *Norton* model is the most common secondary creep model, where the creep rate is assumed proportional to a power law of the equivalent stress σ_e such that

$$f(\sigma_e) = A\left(\frac{\sigma_e}{\sigma_{ref}}\right)^n \tag{3-147}$$

where A is the creep rate coefficient, n is the stress exponent, σ_{ref} is a reference stress level.

At very high stress levels the creep rate is proportional to the exponential of σ_e. Garofalo showed (Ref. 8, Ref. 9) that the power law and exponential creep are limiting cases for the general empirical expression

$$\dot{\varepsilon}_{cr} \propto (\sinh(\alpha\sigma_e))^n$$

This equation reduces to a power law (the Norton model) when $\alpha\sigma_e < 0.8$ and approaches exponential creep for $\alpha\sigma_e > 1.2$, where $1/\alpha$ is a reference equivalent stress level. The *Garofalo (Hyperbolic sine law)* creep model implemented in COMSOL Multiphysics is given as

$$f(\sigma_e) = A\left(\sinh\left(\frac{\sigma_e}{\sigma_{ref}}\right)\right)^n$$

For metals at low stress levels and high temperatures, Nabarro and Herring (Ref. 6, Ref. 7) independently derived an expression for the creep rate as a function of atomic diffusion, so called diffusional creep. The *Nabarro–Herring* model implemented in COMSOL Multiphysics is given as

$$f(\sigma_e) = \frac{7D_v b^3}{k_B T d^2}\sigma_e$$

where d is the grain diameter, D_v is the volume diffusivity through the grain interior, b is the Burgers vector, k_B is the Boltzmann's constant, T is the absolute temperature.

Another model for diffusional creep is the *Coble* creep model (Ref. 6, Ref. 7), which is closely related to Nabarro–Herring creep, but takes into account the grain boundary diffusivity through the parameter D_{gb}. The *Coble* creep model implemented in COMSOL Multiphysics is given as

$$f(\sigma_e) = \frac{50D_{gb} b^4}{k_B T d^3}\sigma_e$$

For metals at intermediate to high stress levels and temperatures close to the melting temperature of the solid, the creep mechanism can be assumed to be diffusion-controlled movements of dislocations in the crystal lattices (Ref. 7). This type of creep deformation can be described by the *Weertman* model, which is given as

$$f(\sigma_e) = \frac{Db}{k_B T}\sigma_{ref}\left(\frac{\sigma_e}{\sigma_{ref}}\right)^n$$

Generally, the stress exponent n in the Weertman model takes a value between 3 and 5 when modeling dislocational creep.

Coble creep is more sensitive to grain diameter than Nabarro–Herring creep.

Equivalent Stress, Creep Flow and Hardening Rule

For metals and alloys, creep is in most cases deviatoric (volume preserving), and can often be described by a *von Mises* equivalent stress

$$\sigma_e = \sqrt{3J_2(\sigma)} = \sqrt{\frac{3}{2}\mathrm{dev}(\sigma):\mathrm{dev}(\sigma)}$$

where J_2 is the second invariant of the stress deviator. If creep is orthotropic, the *Hill orthotropic* equivalent stress can be used, which is given by

$$\sigma_e^2 = \frac{3}{2F+H+G}[F(\sigma_{22}-\sigma_{33})^2 + G(\sigma_{33}-\sigma_{11})^2 + H(\sigma_{11}-\sigma_{22})^2 + 2(L\sigma_{23}^2 + M\sigma_{31}^2 + N\sigma_{12}^2)] \quad (3\text{-}148)$$

where F, G, H, L, M, and N are Hill's coefficients that relate the anisotropy to the local coordinates. In the above equation it is assumed that an average equivalent stress can be computed from F, G, and H using a von Mises like expression; this assumption leads to the first term on the right-hand-side of Equation 3-148.

For soils and other geological materials, creep is often volumetric and can be described by a *Pressure* equivalent stress

$$\sigma_e = |p| = \frac{1}{3}|\mathrm{tr}(\sigma)|$$

where p is the pressure.

It is also possible to specify a user defined equivalent stress in terms of the stress tensor and its invariants.

For more information and the background on the equivalent stress definitions, see Elastoplastic Materials.

Assuming associative creep flow, the creep flow rule is given as function of the equivalent stress

$$\dot{\varepsilon}_{cr} = \lambda\frac{\partial Q_{cr}}{\partial\sigma} = \lambda\frac{\partial\sigma_e}{\partial\sigma}$$

The hardening rule is also assumed to be associative.

The dimensionless tensor N, which gives the direction of the creep flow, is computed from the creep potential Q_{cr} as

$$N = \frac{\partial Q_{cr}}{\partial\sigma}$$

When von Mises equivalent stress is used, the potential equals $Q_{cr} = \sigma_{mises}$, and the deviatoric tensor N_{dev} is defined as done for $J2$ plasticity

$$N_{\text{dev}} = \frac{3}{2}\frac{\text{dev}(\sigma)}{\sigma_e}$$

the creep flow rule is then written as

$$\dot{\varepsilon}_{\text{cr}} = \lambda\frac{\partial Q_{\text{cr}}}{\partial \sigma} = \lambda N_{\text{dev}}$$

Given the property $N_{\text{dev}}{:}N_{\text{dev}} = 3/2$, the evolution of the equivalent creep strain reads

$$\dot{\varepsilon}_{\text{ce}} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{\text{cr}}{:}\dot{\varepsilon}_{\text{cr}}} = |\lambda| \tag{3-149}$$

The same hardening rule is also used when the *Hill orthotropic* equivalent stress is selected.

For the *Pressure* equivalent stress, the flow direction is volumetric

$$N_{\text{vol}} = \frac{\partial Q_{\text{cr}}}{\partial \sigma} = \frac{1}{3}I$$

the volumetric creep flow rule is given as

$$\dot{\varepsilon}_{\text{cr}} = \lambda\frac{\partial Q_{\text{cr}}}{\partial \sigma} = \lambda N_{\text{vol}}$$

and the hardening rule reads

$$\dot{\varepsilon}_{\text{ce}} = \left|\text{tr}(\dot{\varepsilon}_{\text{cr}})\right| = \dot{\varepsilon}_{\text{cr,vol}} = |\lambda|$$

When a user-defined equivalent stress is used, a von Mises like hardening rule is used according to Equation 3-149.

Hardening Function

The creep hardening function h is typically a function of time t and the equivalent creep strain ε_{ce}. In COMSOL Multiphysics, there are two built-in hardening functions, *Strain hardening* and *Time hardening*, and it is also possible to specify a user defined expression. Also, it is possible to add hardening to any of the built-in creep models.

A common model for modeling primary and secondary creep together is the so-called Norton–Bailey (or Bailey–Norton) model. Here, the creep strain is proportional to a power of the equivalent stress (Norton law) and a power of time (Bailey law)

$$\varepsilon_{cr} \propto \sigma_e^n t^m$$

by taking the time derivative of this generic expression at constant stress, it is found that the creep strain rate and stress are related by the *time hardening* function

$$\dot{\varepsilon}_{cr} \propto m\sigma_e^n t^{m-1} = \sigma_e^n h(t)$$

where the *time hardening* function is given by the power law

$$h(t) = m\left(\frac{t + t_{shift}}{t_{ref}}\right)^{m-1}$$

here, t_{shift} is a time shift, t_{ref} is a reference time, and m is the hardening exponent.

Similarly, the *strain hardening* function is given by the power law

$$h(\varepsilon_{ce}) = m\left(\frac{\varepsilon_{ce} + \varepsilon_{shift}}{t_{ref} f(\sigma_e) g(T)}\right)^{\frac{m-1}{m}}$$

where ε_{shift} is the equivalent creep strain shift, t_{ref} is a reference time, and m is the hardening exponent.

The strain shift ε_{shift} and the time shift t_{shift} in the hardening functions serve two purposes:

- To initialize a study where hardening has already taken place.
- To remove singularities. The strain rate expressions actually predict an infinite creep rate at $t = 0$, unless a shift is used. This singularity is weak in the sense that the time integral is well defined, but it will cause numerical problems. Adding a small shift overcomes this problem.

Both the strain hardening function $h(\varepsilon_{ce})$ and the time hardening function $h(t)$ return a dimensionless expression, but for elastoplastic material models with Numerical Integration Algorithm, the Hardening Function returns an expression in units of stress (SI unit: Pa).

Thermal Function

The temperature shift function g typically depends on the absolute temperature T. In COMSOL Multiphysics, an *Arrhenius* temperature function is built-in, but it is also possible to specify a user defined expression.

The *Arrhenius* temperature function is given as

$$g(T) = \exp\left[-\frac{Q}{R}\left(\frac{1}{T}-\frac{1}{T_{\text{ref}}}\right)\right]$$

where Q is the activation energy (J/mol), R is the gas constant, T is the absolute temperature, and T_{ref} is a reference temperature.

Combining Creep Models

Using a combination of f, g, and h together with the definition of the equivalent stress σ_e makes it possible to construct sophisticate creep models. For example, the often used temperature-dependent isotropic Norton creep model is obtained by the following settings:

- **Creep model**: Norton
- **Equivalent stress**: von Mises
- **Thermal effects**: Arrhenius

The resulting creep model is then given by the following equations:

$$\dot{\varepsilon}_{\text{cr}} = \lambda\frac{\partial Q_{\text{cr}}}{\partial \sigma} = A\left(\frac{\sigma_e}{\sigma_{\text{ref}}}\right)^n \exp\left[-\frac{Q}{R}\left(\frac{1}{T}-\frac{1}{T_{\text{ref}}}\right)\right] N_{\text{dev}}$$

Furthermore, selecting a time hardening function results in the Norton–Bailey creep model:

$$\dot{\varepsilon}_{\text{cr}} = \lambda\frac{\partial Q_{\text{cr}}}{\partial \sigma} = A\left(\frac{\sigma_e}{\sigma_{\text{ref}}}\right)^n m\left(\frac{t+t_{\text{shift}}}{t_{\text{ref}}}\right)^{m-1} \exp\left[-\frac{Q}{R}\left(\frac{1}{T}-\frac{1}{T_{\text{ref}}}\right)\right] N_{\text{dev}}$$

Several creep rate contributions can be added to the total creep rate to construct more elaborate models. This is done by adding one or more **Additional Creep** subnodes to an existing **Creep** node. There are no restrictions on how to combine models; for example,

a *von Mises* based creep flow contribution can be combined with a *Pressure* based creep flow contribution.

- In the example *Combining Creep Material Models*:
  Application Library path **Nonlinear_Structural_Materials_Module/Creep/combined_creep**
  a Norton–Bailey model for primary creep is combined with a Norton model for secondary creep. Both creep models are temperature dependent. The resulting creep-rate equation becomes

$$\dot{\varepsilon}_{cr} = A_1\left(\frac{\sigma_e}{\sigma_{ref}}\right)^n m\left(\frac{t+t_{shift}}{t_{ref}}\right)^{m-1}\exp\left[-\frac{Q_1}{R}\left(\frac{1}{T}-\frac{1}{T_1}\right)\right]N_{dev}$$
$$+ A_2\left(\frac{\sigma_e}{\sigma_{ref}}\right)\exp\left[-\frac{Q_2}{R}\left(\frac{1}{T}-\frac{1}{T_2}\right)\right]N_{dev}$$

Time Integration

The rate equations given by the creep flow and hardening rules have to be integrated in time to compute the value of the creep strain tensor ε_{cr} and the equivalent creep strain ε_{ce} at each time step. This can be done using any of the following methods:

- Backward Euler
- Forward Euler
- Domain ODEs

For an arbitrary number of contributions to the creep deformation, the general equations to be solved can be formulated as

$$\dot{\varepsilon}_{cr} = \sum_i\left(\lambda\frac{\partial Q_{cr}}{\partial\sigma}\right)_i = \sum_i(\lambda N_{cr})_i \tag{3-150}$$

$$\dot{\varepsilon}_{ce} = \sum_i(\lambda N_{ce})_i \tag{3-151}$$

where N_{cr} and N_{ce} are tensors that describe the direction of ε_{cr} and ε_{ce}, respectively.

The *Backward Euler* method is used to discretize these equations as

$$\varepsilon_{cr}^{n+1} - \varepsilon_{cr}^{n} = \Delta t \sum_i (\lambda^{n+1} N_{cr}^{n+1})_i \tag{3-152}$$

$$\varepsilon_{ce}^{n+1} - \varepsilon_{ce}^{n} = \Delta t \sum_i (\lambda^{n+1} N_{ce}^{n+1})_i \tag{3-153}$$

where $n+1$ indicates that the variable is evaluated at the current time step, and Δt is the time step. The creep strain tensor ε_{cr}^{n} and equivalent creep strain ε_{ce}^{n} at the previous time step are stored as internal state variables. Equation 3-152 and Equation 3-153 define a system of nonlinear equations that is solved locally at each Gauss point for ε_{cr}^{n+1} and ε_{ce}^{n+1} using Newton's method.

The *Forward Euler* method is used to discretize Equation 3-150 and Equation 3-151 as

$$\varepsilon_{cr}^{n+1} - \varepsilon_{cr}^{n} = \Delta t \sum_i (\lambda N_{cr})_i^n \tag{3-154}$$

$$\varepsilon_{ce}^{n+1} - \varepsilon_{ce}^{n} = \Delta t \sum_i (\lambda N_{ce})_i^n \tag{3-155}$$

such that all quantities on the right-hand-side are evaluated at the previous time step. This means that Equation 3-154 and Equation 3-155 define an explicit update of ε_{cr}^{n+1} and ε_{ce}^{n+1} given only known quantities from the previous time step. The forward Euler method is only conditionally stable, and the time step of the global time marching scheme has to be constrained to preserve stability and accuracy of the method. In COMSOL Multiphysics the limit for a stable time step is estimated as

$$\Delta t_{max} = \min\left(\frac{1}{4} \frac{\|\varepsilon_{el}\|}{\sum_i (\lambda N_{ce})_i^n} \right)$$

where ε_{el} is the elastic strain tensor. This estimate in principle restricts the creep increment to be smaller than the current state of stress, and $1/4$ is a constant to make the estimate conservative. The final value is computed as the minimum value over all Gauss points in the domain where the creep feature is active.

For *Domain ODEs*, Equation 3-150 and Equation 3-151 are converted to weak-form and solved as part of the general initial-boundary value problem. The components of

the creep strain tensor ε_{cr} and the equivalent creep strain ε_{ce} are then treated as degrees-of-freedom.

For more information see the Modeling with ODEs and DAEs and The ODE and DAE Interfaces chapters in the *COMSOL Multiphysics Reference Manual*.

Converting Between Different Data Representations

The equations described in the previous sections about the different creep models differ from the forms most commonly found in the literature. The main difference lies in the introduction of normalizing reference values such as the reference stress σ_{ref} and reference time t_{ref}. These values are in a sense superfluous and can in principle be chosen arbitrarily. The choice of reference values, however, affects the numerical values to be entered for the material data. The implementation in COMSOL Multiphysics has two advantages:

- It is possible to use the built-in unit management system.
- There is not any need to perform any difficult unit conversions when creep data are available in units other than the units in the model. Since many creep models contain the stress or time raised to a noninteger power, such conversions are error prone.

Norton Law

Material data for a Norton law is often available as the parameters A_N and n in the equation

$$\dot{\varepsilon}_{cr} = A_N \sigma_e^{\,n}$$

The coefficient A_N has a physical dimension that depends on the value of n, and the unit has an implicit dependence on the stress and time units. Converting the data to the format used in COMSOL Multiphysics (for example, Equation 3-147) requires the introduction of the reference stress σ_{ref}. It is convenient here to use the implicit stress unit for which A_N is given as the reference stress. The creep rate coefficient A will then have the same numerical value as A_N, and you do not need to do any conversions.

The physical dimension of A is, however, $(\text{time})^{-1}$, whereas the physical dimension of A_N is $(\text{stress})^{-n}(\text{time})^{-1}$.

Another popular way of representing creep data is to supply the stress giving a certain creep rate. As an example, σ_{c7} is the stress at which the creep strain rate is $10^{-7}/\text{h}$. Data on this form is also easy to enter: You set the reference stress σ_{ref} to the value of σ_{c7} and enter the creep rate coefficient A as `1e-7[1/h]`.

Example

Assume that a carbon steel has the following two equivalent descriptions of its creep properties at a certain temperature:

- σ_{c7} = 70 MPa, and stress exponent n = 4.5.
- $A_N = 4.98 \cdot 10^{-16}$ with respect to units MPa and hours, and stress exponent n = 4.5.

In the first case, enter:

- σ_{ref} as `70[MPa]`
- n as 4.5
- A as `1e-7[1/h]`.

In the second case, enter:

- σ_{ref} as `1[MPa]`
- n as 4.5
- A as `4.98e-16[1/h]`

These two sets of data describe the same material.

Garofalo Law

Since the stress inside in the Garofalo law appears as an argument to a `sinh()` function, it must necessarily be dimensionless. Most commonly this is however written as

$$\dot{\varepsilon}_{cr} \propto (\sinh(\alpha\sigma_e))^n$$

Comparing with the expression in COMSOL Multiphysics,

$$\dot{\varepsilon}_{cr} \propto \left(\sinh\left(\frac{\sigma_e}{\sigma_{ref}}\right)\right)^n$$

it is evident that the reference stress should be chosen as

$$\sigma_{ref} = \frac{1}{\alpha}$$

In this case, there is no arbitrariness in the choice of σ_{ref}, since α is an actual material parameter.

VISCOPLASTICITY

Viscoplasticity, as well as Creep, is an inelastic time-dependent deformation that occurs when the material is subjected to stress (typically much less than the yield stress). The growth of unrecoverable strains depend on the rate at which loads are applied, effect that is normally enhanced by high temperatures.

The viscoplastic strain ε_{vp} is computed by a flow rule that defines the relationship between the rate of ε_{vp} and the current state of stress and temperature.

When an associative flow and von Mises equivalent stress σ_e are used, so $Q_{vp} = \sigma_e$, the rate of ε_{vp} is coaxial to the stress deviator and the viscoplastic flow rule is written as

$$\dot{\varepsilon}_{vp} = \lambda_{vp} \frac{\partial Q_{vp}}{\partial \sigma} = \lambda_{vp} \frac{3}{2} \frac{\mathrm{dev}(\sigma)}{\sigma_e} = \lambda_{vp} N_{dev}$$

where Q_{vp} is the viscoplastic potential, σ is the stress tensor, and $N_{dev} = \partial Q_{vp} / \partial \sigma$ is the flow direction.

The viscoplastic multiplier λ_{vp} depicts a different expression for different viscoplastic models (Anand, Chaboche, and so on), and it is equal to the equivalent viscoplastic strain rate $\dot{\varepsilon}_{vpe}$

$$\dot{\varepsilon}_{vpe} = \sqrt{\frac{2}{3} \dot{\varepsilon}_{vp} : \dot{\varepsilon}_{vp}} = \sqrt{\frac{2}{3} \lambda_{vp}^2 N_{dev} : N_{dev}} = |\lambda_{vp}|$$

given the property $N_{dev} : N_{dev} = 3/2$.

Setting different measures for the equivalent stress σ_e (von Mises, Tresca, Hill) allows to define different viscoplastic multiplier and flow directions, see Equivalent Stress, Creep Flow and Hardening Rule for details.

Anand Model

Anand viscoplasticity (Ref. 9) is a deviatoric viscoplastic model suitable for isotropic viscoplastic deformations. As for the models described in Creep, the Anand model has no yield surface.

The rate of ε_{vp} is determined by the viscoplastic multiplier λ_{vp} which is a function of stress σ and temperature T, and it is given by

$$\lambda_{vp} = Ae^{-Q/RT}\left[\sinh\left(\xi\frac{\sigma_e}{s_a}\right)\right]^{\frac{1}{m}}$$

here, A is the viscoplastic rate coefficient, Q is the activation energy (J/mol), ξ is the stress multiplier, m is the stress sensitivity, and R is the gas constant.

The *deformation resistance*, s_a (SI unit: Pa), controls the hardening behavior of the Anand model. In COMSOL Multiphysics it is derived for the normalized *resistance factor*, s_f, and the *deformation resistance saturation coefficient*, s_{sat}, such that $s_a = s_{sat}\, s_f$. The resistance factor s_f is a dimensionless internal variable that is computed by the evolution equation

$$\dot{s}_f = h\lambda_{vp}$$

with the initial condition $s_f(0) = s_{init} / s_{sat}$. The evolution of the resistance factor s_f is controlled by the dimensionless hardening function

$$h = \frac{h_0}{s_{sat}}\left|1 - \frac{s_f}{s_f^*}\right|^a \operatorname{sign}\left(1 - \frac{s_f}{s_f^*}\right) = \frac{h_0}{s_{sat}}\left|1 - \frac{s_f}{s_f^*}\right|^{a-1}\left(1 - \frac{s_f}{s_f^*}\right)$$

were h_0 (SI unit: Pa) represents a constant rate of *athermal hardening* in the curve stress versus strain (Ref. 9). The use of the sign function and the absolute value in the hardening function permits the modeling of either strain hardening or strain softening, depending on whether s_f is greater or smaller than the saturation value s_f^*.

The saturation value s_f^* is calculated from the expression

$$s_f^* = \left(\frac{\lambda_{vp}}{A}e^{Q/RT}\right)^n$$

where n is the *deformation resistance sensitivity* exponent.

The Anand model computes the equivalent viscoplastic strain rate (viscoplastic multiplier) using the evolution equation for the internal variable

$$\dot{\varepsilon}_{vpe} = \lambda_{vp} = Ae^{-Q/RT}\left[\sinh\left(\xi\frac{\sigma_e}{s_a}\right)\right]^{\frac{1}{m}}$$

Anand–Narayan Model

Anand–Narayan model (Ref. 10) is suitable for modeling the viscoplastic deformation of lithium.

The equivalent viscoplastic strain rate (viscoplastic multiplier) is given by

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = Ae^{-Q/RT}\left(\xi\frac{\sigma_e}{s_a}\right)^{\frac{1}{m}}$$

where A is the viscoplastic rate coefficient, Q is the activation energy (J/mol), ξ is the stress multiplier, m is the stress sensitivity, and R is the gas constant.

The *deformation resistance*, s_a (SI unit: Pa), is computed with an extra dependent variable as done for Anand Model.

Chaboche Model

The equivalent viscoplastic strain rate (viscoplastic multiplier) is given by

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = A\left\langle\frac{\sigma_e - \sigma_{ys}}{\sigma_{ref}}\right\rangle^n \quad \text{for } \sigma_e - \sigma_{ys} \geq 0$$

and the viscoplastic strain rate tensor is given by

$$\dot{\varepsilon}_{vp} = \lambda_{vp} N_{dev}$$

Here, A is the viscoplastic rate coefficient (SI unit: 1/s), n is the stress exponent (dimensionless), and σ_{ref} is a reference stress level (SI unit: Pa).

The Macaulay brackets $\langle \cdot \rangle$ are applied on the yield function, which is defined as done for plasticity

$$F = \sigma_e - \sigma_{ys}$$

The equivalent stress σ_e is either the von Mises, Tresca, or Hill orthotropic stress, or a user-defined expression; and σ_{ys} is the yield stress (which may include an Numerical Integration Algorithm model). The stress tensor used in the equivalent stress σ_e is shifted by what is usually called the *back stress*, σ_{back} when Kinematic Hardening is included.

The deviatoric tensor N_{dev} is computed from the viscoplastic potential Q_{vp}

$$N_{dev} = \frac{\partial Q_{vp}}{\partial \sigma}$$

When von Mises equivalent stress is used, the associated flow rule reads $Q_{vp} = F$, and the deviatoric tensor N_{dev} is defined as done for deviatoric creep.

	For more information see Converting Between Different Data Representationsthe for the relation between the viscoplastic rate coefficient A, the stress exponent n, and the reference stress σ_{ref}.

Perzyna Model

The Perzyna model defines the equivalent viscoplastic strain rate (viscoplastic multiplier) by

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = A\left\langle \frac{\sigma_e}{\sigma_{ys}} - 1 \right\rangle^n \text{ for } F = \sigma_e - \sigma_{ys} \geq 0$$

and the viscoplastic strain rate tensor is given by

$$\dot{\varepsilon}_{vp} = \lambda_{vp} N_{dev}$$

As described for Chaboche Model, the equivalent stress σ_e can be either von Mises (default), Tresca, Hill orthotropic stress, or a user defined expression. The flow direction $N_{dev} = \partial Q_{vp} / \partial \sigma$ is computed from the viscoplastic potential Q_{vp}, which can be associated or nonassociated.

Bingham Model

Bingham model is equivalent to Chaboche Model when setting the exponent $n = 1$, and denoting the viscosity as the quotient $\eta = \sigma_{ref}/A$. The equivalent viscoplastic strain rate (viscoplastic multiplier) is given by

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = \frac{1}{\eta}\left\langle \sigma_e - \sigma_{ys} \right\rangle \text{ for } F = \sigma_e - \sigma_{ys} \geq 0$$

and the viscoplastic strain rate tensor is given by

$$\dot{\varepsilon}_{vp} = \lambda_{vp} N_{dev}$$

As described for Chaboche Model, the equivalent stress σ_e can be either von Mises (default), Tresca, Hill orthotropic stress, or a user defined expression. The flow direction $N_{dev} = \partial Q_{vp} / \partial \sigma$ is computed from the viscoplastic potential Q_{vp}, which can be associated or nonassociated.

Peric Model

The Peric model defines the equivalent viscoplastic strain rate (viscoplastic multiplier) by

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = A\langle\left(\frac{\sigma_e}{\sigma_{ys}}\right)^n - 1\rangle \text{ for } F = \sigma_e - \sigma_{ys} \geq 0$$

and the viscoplastic strain rate tensor is given by

$$\dot{\varepsilon}_{vp} = \lambda_{vp} N_{dev}$$

As described for Chaboche Model, the equivalent stress σ_e can be either von Mises (default), Tresca, Hill orthotropic stress, or a user defined expression. The flow direction $N_{dev} = \partial Q_{vp} / \partial\sigma$ is computed from the viscoplastic potential Q_{vp}, which can be associated or nonassociated.

User Defined

It is possible to freely define the viscoplastic rate λ_{vp}

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} \text{ for } F = \sigma_e - \sigma_{ys} \geq 0$$

The conditional evaluation $F \geq 0$ is handled automatically.

By default, an associative flow and von Mises equivalent stress σ_e are used, so $Q_{vp} = F$, the rate of ε_{vp} is coaxial to the stress deviator, and the viscoplastic flow rule is written as

$$\dot{\varepsilon}_{vp} = \lambda_{vp}\frac{\partial Q_{vp}}{\partial\sigma} = \lambda_{vp}\frac{3}{2}\frac{\text{dev}(\sigma)}{\sigma_e} = \lambda_{vp} N_{dev}$$

Thermal Effects

The temperature function g typically depends on the absolute temperature T. In COMSOL Multiphysics, an *Arrhenius* temperature function is built-in, but it is also possible to specify a user defined expression.

The *Arrhenius* temperature function is given as

$$g(T) = \exp\left[-\frac{Q}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right]$$

where Q is the activation energy (J/mol), R is the gas constant, T is the absolute temperature, and T_{ref} is a reference temperature.

	See also the description of Viscoplasticity in the physics interface documentation.

Time Stepping

The rate equations given by the viscoplastic flow rule and evolution equations for the two internal variables are integrated in time to compute the value of the viscoplastic strain tensor ε_{vp}, the equivalent creep strain ε_{vpe}, and other possible internal variables (for instance, the resistance factor s_{f} in Anand model) at each time step. This can be done using any of the following methods:

- Backward Euler
- Domain ODEs

The *Backward Euler* method is used to discretize these equations as

$$\varepsilon_{\text{vp}}^{n+1} - \varepsilon_{\text{vp}}^{n} = \Delta t \lambda_{\text{vp}}^{n+1} N_{\text{vp}}^{n+1} \tag{3-156}$$

$$\varepsilon_{\text{vpe}}^{n+1} - \varepsilon_{\text{vpe}}^{n} = \Delta t \lambda_{\text{vp}}^{n+1} N_{\text{vpe}}^{n+1} \tag{3-157}$$

where N_{vp} and N_{vpe} are tensors that describe the direction of ε_{vp} and ε_{vpe}, respectively, $n+1$ indicates that the variable is evaluated at the current time step, and Δt is the time step.

For Anand Model an extra equation is solved for the *resistance factor*, s_{f},

$$s_{\text{f}}^{n+1} - s_{\text{f}}^{n} = \Delta t \lambda_{\text{vp}}^{n+1} h^{n+1} \tag{3-158}$$

The viscoplastic strain tensor $\varepsilon_{\text{vp}}^{n}$, equivalent viscoplastic strain $\varepsilon_{\text{vpe}}^{n}$, and the resistance factor s_{f}^{n} at the previous time step are stored as internal state variables. Equation 3-156 to Equation 3-158 define a system of nonlinear equations that is solved locally at each Gauss point for $\varepsilon_{\text{vp}}^{n+1}$, $\varepsilon_{\text{vp}}^{n+1}$, and s_{f}^{n+1} using Newton's method.

For *Domain ODEs*, the evolution equations of the viscoplastic model are converted to weak-form and solved as part of the general initial-boundary value problem. The components of ε_{vp}, ε_{vpe}, and s_f are then treated as degrees of freedom.

For more information see the Modeling with ODEs and DAEs and The ODE and DAE Interfaces chapters in the *COMSOL Multiphysics Reference Manual.*

CREEP AND VISCOPLASTICITY FOR LARGE STRAINS

When the **Large strains** formulation is selected in the Creep or Viscoplasticity nodes, a Multiplicative Decomposition of deformation gradients is used

$$F = F_{el}F_{in}$$

here, F is the deformation gradient, F_{el} is the elastic deformation, and F_{in} is inelastic deformation due to creep or viscoplasticity.

The velocity gradient then reads (see Strain Rate and Spin for details)

$$L = \dot{F}F^{-1} = (\dot{F}_{el}F_{in} + F_{el}\dot{F}_{in})F^{-1} = \dot{F}_{el}F_{el} + FF_{in}^{-1}\dot{F}_{in}F^{-1}$$

The velocity gradient can then be split into the elastic and inelastic parts, $L = L_{el} + L_{in}$. The inelastic velocity gradient

$$L_{in} = FF_{in}^{-1}\dot{F}_{in}F^{-1}$$

is energy conjugate to the stress tensor, and it is coaxial to it for maximum dissipation. Neglecting the inelastic spin, so the inelastic deformation gradient is symmetric, the formulation for large strain viscoplasticity or creep is similar to the formulation described in the Multiplicative Plastic Flow section. The time-derivative of the inverse inelastic deformation gradient is given by the rate

$$\dot{F}_{in}^{-1} = -\lambda_{in}F^{-1}\frac{\partial Q_{in}}{\partial \sigma}FF_{in}^{-1} \qquad (3\text{-}159)$$

where λ_{in} is the equivalent creep or viscoplastic strain rate, and Q_{in} is the creep or viscoplastic potential. For creep and viscoplasticity in metals (so-called *J2 plasticity*) the viscoplastic potential depends on the deviatoric stress, $Q_{in} = Q_{in}(J_2(\sigma))$, and the flow direction is deviatoric, $N_{dev} = \partial Q_{in} / \partial \sigma$.

For large strains, Equation 3-159 is numerically solved with the so-called *exponential mapping* technique, as described in Numerical Integration Algorithm, and the elastic deformation gradient then obtained from

$$\boldsymbol{F}_{\text{el}} = \boldsymbol{F}\boldsymbol{F}_{\text{in}}^{-1}$$

POLYMER VISCOPLASTICITY

Elastomers and biological materials exhibit strain rate dependent mechanical behavior and hysteresis when subjected to cyclical loads

Bergstrom–Boyce Model

The Bergstrom–Boyce model (Ref. 35–36) is a viscoplastic material that successfully capture such phenomena, and it is suitable for modeling rubbers and elastomers.

The rheology of this material model is similar to the Standard Linear Solid Model, but the deformation in the spring–dashpot branch is nonlinear.

The stiffness in the spring is defined by the energy factor β_v. This factor multiplies the isochoric strain energy density defined in the main hyperelastic branch, in a similar way as done for Large Strain Viscoelasticity. For instance, if the main branch is defined by an Arruda–Boyce hyperelastic material with a shear modulus μ_0, then the shear modulus in the spring is $\mu_1 = \beta_v\mu_0$.

The rate of viscoplastic deformation is given by a Multiplicative Decomposition of deformation gradients

$$\boldsymbol{F} = \boldsymbol{F}_{\text{el}}\boldsymbol{F}_{\text{vp}}$$

here, $\boldsymbol{F}$ is the deformation gradient in the hyperelastic branch, $\boldsymbol{F}_{\text{el}}$ is the elastic deformation in the spring, and $\boldsymbol{F}_{\text{vp}}$ is inelastic deformation in the dashpot.

The velocity gradient reads (see Creep and Viscoplasticity for Large Strains)

$$\boldsymbol{L} = \dot{\boldsymbol{F}}\boldsymbol{F}^{-1} = (\dot{\boldsymbol{F}}_{\text{el}}\boldsymbol{F}_{\text{vp}} + \boldsymbol{F}_{\text{el}}\dot{\boldsymbol{F}}_{\text{vp}})\boldsymbol{F}^{-1} = \dot{\boldsymbol{F}}_{\text{el}}\boldsymbol{F}_{\text{el}} + \boldsymbol{F}\boldsymbol{F}_{\text{vp}}^{-1}\dot{\boldsymbol{F}}_{\text{vp}}\boldsymbol{F}^{-1}$$

The viscoplastic velocity gradient, $\boldsymbol{L}_{\text{vp}}$, is energy conjugate to the stress tensor σ in the branch. For volume-preserving viscoplasticity, it is defined as

$$\boldsymbol{L}_{\text{vp}} = \boldsymbol{F}\boldsymbol{F}_{\text{vp}}^{-1}\dot{\boldsymbol{F}}_{\text{vp}}\boldsymbol{F}^{-1} = \lambda_{\text{vp}}\boldsymbol{N}_{\text{dev}} \text{ with } \boldsymbol{N}_{\text{dev}} = \frac{3}{2}\frac{\text{dev}(\sigma)}{\sigma_e}$$

The equivalent viscoplastic strain rate (viscoplastic multiplier) is similar to the rate defined for the Chaboche Model, but it also incorporates strain hardening

$$\lambda_{vp} = \dot{\varepsilon}_{vpe} = Ah(\lambda_{vpe})\left\langle \frac{\sigma_e - \sigma_{co}}{\sigma_{res}} \right\rangle^n \quad \text{for } \sigma_e - \sigma_{co} \geq 0$$

here, A is the viscoplastic rate coefficient, σ_e is the von Mises stress in the network, σ_{co} is a cutoff stress, σ_{res} is the flow resistance, and n is the stress exponent.

For more information see Converting Between Different Data Representations the for the relation between the viscoplastic rate coefficient A, the stress exponent n, and the flow resistance σ_{res}.

The strain hardening function $h(\lambda_{vpe})$ is defined by the power law

$$h(\lambda_{vpe}) = (\lambda_{vpe} - 1 + \xi)^c$$

Here, c is the strain hardening exponent and ξ is a small numerical constant. The equivalent viscoplastic stretch λ_{vpe} is computed from the viscoplastic deformation

$$\lambda_{vpe} = \sqrt{\frac{\boldsymbol{F}_{vp} : \boldsymbol{F}_{vp}}{3}}$$

Bergstrom–Bischoff Model

The Bergstrom–Bischoff model, also known as the *three network model* (Ref. 35, Ref. 37) is suitable for modeling elastomers and thermoplastics. The rheology is similar to the Bergstrom–Boyce model, but it adds two parallel branches instead of one.

The stiffness in the spring in the first branch is defined by the energy factor β_{v1}. This factor multiplies the isochoric strain energy density defined in the main hyperelastic branch, in a similar way as done for Bergstrom–Boyce Model. For instance, if the main branch is defined by an Arruda–Boyce hyperelastic material with a shear modulus μ_0, then the stiffness in the spring is $\mu_1 = \beta_{v1}\mu_0$.

The viscoplastic rate in the first branch includes a power law to represent stress hardening or softening

$$\lambda_{vp1} = \dot{\varepsilon}_{vpe1} = A\left(\frac{\sigma_{e,1}}{\sigma_{res,1} + a_1 p}\right)^{n_1}$$

here, A is the viscoplastic rate coefficient, $\sigma_{e,1}$ is the von Mises stress in the branch, $\sigma_{res,1}$ is the flow resistance, a_1 is a material parameter to control the influence of the hydrostatic stress, and n_1 is the stress exponent.

The energy factor in the second branch β_{v2} is computed after solving the rate equation

$$\dot{\beta}_{v2} = -\alpha(\beta_{v2} - \beta_{vf})\lambda_{vp1}$$

Here, β_{v2} is the final energy factor after saturation. The energy factor is initialized to a value equal to $\beta_{v2} = \beta_{vi}$. The energy factor β_{v2} then multiplies the isochoric strain energy density defined in the main hyperelastic branch to define the stiffness of the spring in the second branch, in a similar way as done in the first branch.

The viscoplastic rate in the second branch also includes a power law to represent stress hardening or softening

$$\lambda_{vp2} = \dot{\varepsilon}_{vpe2} = A\left(\frac{\sigma_{e,2}}{\sigma_{res,2} + a_2 p}\right)^{n_2}$$

here, $\sigma_{e,2}$ is the von Mises stress in the branch, $\sigma_{res,2}$ is the flow resistance, a_2 is a material parameter to control the influence of the hydrostatic stress, and n_2 is the stress exponent.

The material model also includes a power law expression to model thermal softening, see Thermal Effects for details.

Parallel Network Model

The Parallel Network model (Ref. 35, Ref. 37) can be seen as a generalized version of the Bergstrom–Boyce Model, where several viscoplastic branches can be added in parallel to the main hyperelastic branch.

It is possible to choose a different elastic strain energy for the viscoplastic branch as compared to that of the main hyperelastic branch, however, all the additional viscoplastic branches are assumed to have the same type of hyperelastic strain energy density as the first branch.

For example, it is possible to combine a Neo-Hookean material in the main hyperelastic branch with two parallel spring–dashpot branches characterized by an Arruda–Boyce model.

Several networks can be added to build more elaborate viscoplasticity models, and the energy factors β_{vi} are used to differentiate the response of the additional stiffnesses. This is done by adding one or more **Additional Network** subnodes to an existing **Polymer**

Viscoplasticity node. The rheological representation of this model can be seen in Figure 3-32.

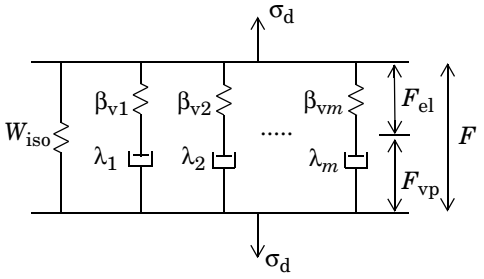


Figure 3-32: Parallel network model.

The dashpots in the additional viscoplastic branches can showcase strain hardening, like in the Bergstrom–Boyce Model, or pressure-dependent stress hardening like in the Bergstrom–Bischoff Model.

User Defined

Enter the equivalent viscoplastic strain rate (viscoplastic multiplier). This can be any expression of the von Mises stress, σ_e, and the equivalent viscoplastic stretch λ_{vpe}

$$\lambda_{vp} = \dot{\varepsilon}_{vpe}(\sigma_e, \lambda_{vpe})$$

The viscoplastic velocity gradient, L_{vp}, is then defined as

$$L_{vp} = FF_{vp}^{-1}\dot{F}_{vp}F^{-1} = \lambda_{vp}N_{dev} \text{ with } N_{dev} = \frac{3}{2}\frac{\text{dev}(\sigma)}{\sigma_e}$$

Thermal Effects

The temperature function $g(T)$ depends on the absolute temperature T, and it acts as a multiplier for the viscoplastic rate.

The *Arrhenius* temperature function is given as

$$g(T) = \exp\left[-\frac{Q}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right]$$

where Q is the activation energy (J/mol), R is the gas constant, T is the absolute temperature, and T_{ref} is a reference temperature.

The *Power law* temperature function is given as

$$g(T) = \left(\frac{T}{T_{\text{ref}}}\right)^m$$

where T is the absolute temperature, T_{ref} is a reference temperature, and m is the temperature exponent.

It is also possible to specify a user defined expression for the temperature function $g(T)$.

ENERGY DISSIPATION

Since creep and viscoplasticity are inelastic processes, the dissipated energy density can be calculated by integrating the *creep dissipation rate density* or *viscoplastic dissipation rate density* given by

$$\dot{W}_{\text{cr}} = \sigma{:}\dot{\varepsilon}_{\text{cr}}$$

$$\dot{W}_{\text{vp}} = \sigma{:}\dot{\varepsilon}_{\text{vp}}$$

in time. The total energy dissipated in a given volume can be calculated by a volume integration of the dissipated creep energy density W_{cr} or dissipated viscoplastic energy density W_{vp}.

When the **Calculate dissipated energy** checkbox is selected, the dissipation rate density due to creep is available under the variable `solid.Wcdr` and the dissipation rate density due to viscoplasticity is available under the variable `solid.Wvpdr`. The dissipated energy density due to creep is available under the variable `solid.Wc` and due to viscoplasticity under the variable `solid.Wvp`. Here `solid` denotes the name of the physics interface node.

Inelastic Strain Rate

The inelastic behavior of materials is often described by the rate of the inelastic quantity, especially if the behavior is time dependent and nonlinear. For example, Elastoplastic Materials, Linear Viscoelasticity, Creep and Viscoplasticity all describe material models that are implemented in dedicated features. The **Inelastic Strain Rate** node provides functionality to specify a general expression for the rate of inelastic deformation, entered either as strain tensor ε_{ie}; deformation gradient F_{ie}; deformation

gradient inverse F_{ie}^{-1}; or stretch λ_{ie}. The user defined expression can be a function of any variable in the model.

Given a generic inelastic quantity U_{ie}, the feature sets up and solves the following type of distributed ODE

$$\frac{dU_{ie}}{dt} = f(t, u, U_{ie}, \ldots) \tag{3-160}$$

where t is the time, u is the displacement, and $f(t, u, U_{ie},...)$ is a user defined expression. Equation 3-160 is also subjected to the initial condition

$$U_{ie}(t_0) = U_{ie}^{init}$$

where U_{ie}^{init} is a user defined input.

Time Integration

The distributed ODE in Equation 3-160 is integrated in time to compute the value of the inelastic quantity U_{ie} at each time step. This can be done using any of the following methods:

- Backward Euler
- Domain ODEs

The *Backward Euler* method is used to discretize Equation 3-160 as

$$\frac{U_{ie}^{n+1} - U_{ie}^{n}}{\Delta t} = f(t^{n+1}, u^{n+1}, U_{ie}^{n+1}, \ldots) \tag{3-161}$$

where n+1 indicates the current time step, and Δt is the time step. The inelastic quantity at the previous time step U_{ie}^{n} is stored as internal state variables, their initial value is given by U_{ie}^{init}. Equation 3-161 defines a system of nonlinear equations that is solved locally at each Gauss point for U_{ie}^{n+1} using Newton's method.

For *Domain ODEs*, Equation 3-160 is converted to a weak form and solved as part of the general initial boundary value problem. The components of the inelastic quantity U_{ie} are here treated as a degrees-of-freedom of the model.

For more information see the Modeling with ODEs and DAEs and The ODE and DAE Interfaces chapters in the *COMSOL Multiphysics Reference Manual.*

Shape Memory Alloy

The term *shape memory alloy* (SMA) normally refers to alloys that can undergo large strains, effect called *superelasticity* or *pseudoelasticity*; and also to materials that after undergoing large deformations can remember their original shape when heated above a certain temperature, so-called *memory* effect.

These alloys are mixtures of metals such as copper, aluminum, nickel, titanium, and other. Nickel-titanium (NiTi) alloys have applications in many industries due to their thermal and mechanical properties. Most of the industrial applications of NiTi alloys take advantage of the different mechanical properties of the two crystalline structures found around room temperature: *austenite* and *martensite*.

Hot NiTi alloys are composed by a pure austenite phase. The martensite phase develops upon cooling the alloy below the *martensite start temperature*, M_s. The martensite volume fraction ξ_M will increase until the cooling temperature reaches the *martensite finish temperature*, M_f, below which the alloy microstructure will be pure martensite.

The reverse process has different transition temperatures: A pure martensite alloy will develop an austenite microstructure if it is heated above the *austenite start temperature*, A_s. The austenite phase will increase upon heating the material to the *austenite finish temperature*, A_f, above which the alloy microstructure becomes 100% austenite.

Many industrial applications take advantage of this hysteresis loop, as the transition temperatures are not the same in a heating-cooling cycle.

Under stress free conditions, these phase transformation temperatures (also called *critical temperatures*) are such that $M_f < M_s < A_s < A_f$.

There are two shape memory alloy models available with the Nonlinear Structural Material Module: the Souza–Auricchio model and the Lagoudas model. These material models differ in the expression for the free energy density.

SOUZA–AURICCHIO MODEL

For the Souza–Auricchio model, Helmholtz free energy density depends on two state variables: the total strain tensor ε and the temperature T. An additional internal variable is used to compute the *transformation strain* tensor ε_{tr} (Ref. 38-40)

$$\psi(\varepsilon, T, \varepsilon_{tr}) = \rho c T(1 - \ln(T)) + \frac{K}{2}\varepsilon_{vol}^2 + G(\varepsilon - \varepsilon_{tr})^2 + \frac{H_k}{2}(\varepsilon_{tr})^2 + \tau_m(T)|\varepsilon_{tr}| + I(\varepsilon_{tr})$$

Here, c is the heat capacity at constant pressure, K and G are the bulk and shear moduli, ε_{vol} is the volumetric strain, H_k is the hardening modulus, and $I(\varepsilon_{tr})$ is the *indicator function* for the strain limit constraint. The *equivalent transformation strain* ε_{tre} is used as a measure of the transformation strain tensor

$$\varepsilon_{tre} = |\varepsilon_{tr}| = \sqrt{\frac{2}{3}\varepsilon_{tr}:\varepsilon_{tr}}$$

The indicator function is defined by

$$I(\varepsilon_{tr}) = \begin{cases} 0 \text{ for } \varepsilon_{tre} < \varepsilon_{tr,\,max} \\ \infty \qquad \text{otherwise} \end{cases}$$

The indicator function plays the role of a penalty function for the equivalent transformation strain ε_{tre}.

The term $\psi_{ch} = \tau_m(T)|\varepsilon_{tr}|$ is denoted as the chemical energy density due to the thermally induced martensite transformation. The function $\tau_m(T)$ corresponds to the temperature-dependent martensite to austenite equilibrium stress, defined from the slope of the limit curve β and the reference temperature T^*

$$\tau_m(T) = \beta\langle T - T^* \rangle$$

Here, the operator $< \cdot >$ denotes the Macaulay brackets.

The conjugated thermodynamic stress σ_{tr} associated to the transformation strain variable is

$$\sigma_{tr} = \frac{\partial \psi}{\partial \varepsilon_{tr}} = \text{dev}(\sigma) - H_k\varepsilon_{tr} - \tau_m(T)\frac{\partial|\varepsilon_{tr}|}{\partial \varepsilon_{tr}} - \frac{\partial I}{\partial \varepsilon_{tr}} \tag{3-162}$$

The evolution of the transformation strain ε_{tr} is given by the so-called *limit function*, which takes the same form as the yield function for von Mises plasticity.

$$F_y = |\sigma_{tr}| - \sigma_0$$

where σ_0 is the *elastic domain radius.*

In Ref. 39 it was suggested to consider the asymmetric behavior that many alloys show in tension and compression. In this case, the *limit function* depends on the Lode angle θ, the elastic radius in tension σ_t, and elastic radius in compression σ_c.

$$F_y = |\sigma_{tr}| + |\sigma_{tr}|\frac{\sigma_c - \sigma_t}{\sigma_c + \sigma_t}\cos(3\theta) - \frac{2\sigma_c\sigma_t}{\sigma_c + \sigma_t}$$

Denoting the *elastic domain radius* in tension $\sigma_0 = \sigma_t$, and the compression to tension stress ratio k such as $\sigma_c = k\sigma_t = k\sigma_0$, the asymmetric *limit function* reads

$$F_y = |\sigma_{tr}| + |\sigma_{tr}|\frac{k-1}{k+1}\cos(3\theta) - \frac{2k}{k+1}\sigma_0$$

In Ref. 39, the stability for the limit function F_y is achieved when $k \leq 1.286$.

For an additive strain decomposition, the evolution equation for the transformation strain ε_{tr} is computed from the flow rule

$$\dot{\varepsilon}_{tr} = \lambda_p \frac{\partial F_y}{\partial \sigma}$$

where the plastic multiplier λ_p is solved with the Kuhn–Tucker conditions, as done for plasticity, see Additive Plastic Flow.

When a multiplicative decomposition is used for the Souza–Auricchio model, an evolution equation is formulated for the transformation deformation gradient $\boldsymbol{F}_{tr}$ in terms of the flow rule

$$\dot{\overline{\boldsymbol{F}_{tr}^{-1}}} = -\lambda_p \boldsymbol{F}^{-1}\frac{\partial F_y}{\partial \sigma}\boldsymbol{F}\boldsymbol{F}_{tr}^{-1}$$

where $\boldsymbol{F}$ is the total deformation gradient. See also Multiplicative Plastic Flow. The last three terms in Equation 3-162 can, moreover, be recognized as the back stress in a linear kinematic hardening model. Hence the framework for kinematic hardening with Multiplicative Decomposition is used.

LAGOUDAS MODEL

For Lagoudas model, Gibbs free energy density depends on two state variables: the total stress tensor σ and the temperature field T. Additional internal variables are used to compute the transformation strain tensor ε_{tr} and the martensite volume fraction ξ (Ref. 41)

$$G(\sigma, T, \varepsilon_{tr}, \xi) = c\left((T - T_0) - T\ln\frac{T}{T_0}\right) - s_0 T - \frac{1}{2\rho}\sigma{:}S{:}\sigma - \frac{1}{\rho}\sigma{:}(\varepsilon_{th} - \varepsilon_{tr}) + u_0 + \frac{1}{\rho}f(\xi)$$

Here, c is the heat capacity at constant pressure, s_0 is the specific entropy at reference state, S is the compliance matrix, ε_{th} is the thermal strain tensor, ε_{tr} is the transformation strain tensor, u_0 is the specific internal energy at reference state, and

$f(\xi)$ is the *transformation hardening function*. The compliance matrix S, is obtained by a volume average of the elastic properties of martensite and austenite

$$S(\xi) = \xi S_M + (1-\xi)S_A = S_A + \xi \Delta S$$

where $\Delta S = S_M - S_A$. Also, other material parameters are averaged this way.

As opposed to Souza–Auricchio model, the evolution equation for the transformation strain ε_{tr} is computed from the flow rule

$$\dot{\varepsilon}_{tr} = \dot{\xi}\Lambda$$

where the normalized *transformation tensor* Λ changes principal directions depending on the direction of the martensitic transformation.

The maximum transformation strain in Lagoudas model can be considered constant, or stress-dependent as described in Ref. 42. A stress-dependent maximum transformation strain can be used at low stress levels, where the martensite turns into detwinned structures.

The variable for the direction of the martensitic transformation, $\dot{\xi}$, is calculated from the previous state to determine the expected increment or decrement of the transformation strain tensor ε_{tr}. This calculation is computational expensive and it can lead to convergence issues. Since in many applications the transformation direction $\dot{\xi}$ is known a priori (for instance, mechanical loading or unloading, or temperature increment/decrement) a user input enables to set the transformation direction manually to 1 or –1, thus speeding up the computational time.

For Lagoudas model, it is possible to choose from different transformation hardening functions $f(\xi)$

- Quadratic
- Cosine
- Smooth
- User defined

The quadratic hardening function is defined as

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b_M \xi^2 + (\mu_1 + \mu_2)\xi & \text{for } \dot{\xi} > 0 \\ \frac{1}{2}\rho b_A \xi^2 + (\mu_1 - \mu_2)\xi & \text{for } \dot{\xi} < 0 \end{cases}$$

where the parameters b_M and b_A are computed from

$$b_M = \Delta s_0(M_f - M_s) \text{ and } b_A = \Delta s_0(A_f - A_s),$$

and the parameters μ_1 and μ_2 from

$$\mu_1 = \frac{1}{2}\rho\Delta s_0(M_s + A_f) - \rho\Delta u_0 \text{ and } \mu_2 = \frac{1}{4}\rho\Delta s_0(A_s - A_f - M_f + M_s)$$

The smooth hardening function is defined with four smoothing parameters n_1, n_2, n_3, and n_4:

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b_M\left(\xi + \frac{\xi^{n_1+1}}{n_1+1} + \frac{(1-\xi)^{n_2+1}}{n_2+1}\right) & \text{for } \dot{\xi} > 0 \\ \frac{1}{2}\rho b_A\left(\xi + \frac{\xi^{n_3+1}}{n_3+1} + \frac{(1-\xi)^{n_4+1}}{n_4+1}\right) & \text{for } \dot{\xi} < 0 \end{cases}$$

EQUIVALENT MATERIAL PROPERTIES FOR LAGOUDAS MODEL

In Lagoudas model, the phase transformation is described by six parameters: the transition temperatures M_f, M_s, A_s, A_f, and the two slopes of austenite and martensite limit curves C_A and C_M.

When the experimental data consists of a uniaxial stress–strain curve taken at a constant temperature T_σ above the austenite finish temperature A_f, it is possible to observe the *pseudoelastic* or *superelastic* effect. In this case it is convenient to enter the transition stress levels instead of the transition temperatures since these are easily derived from the uniaxial curve.

Consider a NiTi alloy composed by 100% of austenite at room temperature $T_\sigma > A_f$. Upon loading, the slope in the stress–strain curve would be the Young's modulus of austenite, E_A. The martensite phase starts to develop when the axial stress reaches the *martensite start stress*, σ_{Ms}. If the alloy is further loaded, the slope reduces, entering a region called *loading plateau*. Above the *martensite finish stress*, σ_{Mf}, the microstructure becomes 100% martensite. This process is commonly called the *forward transformation*.

The reverse process has different transition stresses: When the 100% martensite alloy is unloaded at constant temperature $T_\sigma > A_f$, the slope follows the martensite Young's modulus E_M. When the axial stress reaches the *austenite start stress*, σ_{As}, the austenite volume fraction ξ_A starts to develop until the axial stress falls below the *austenite*

finish stress, σ_{Af}, at which level the alloy microstructure is 100% austenite. This process is commonly called the *reverse transformation*. Figure 3-33 illustrates this process.

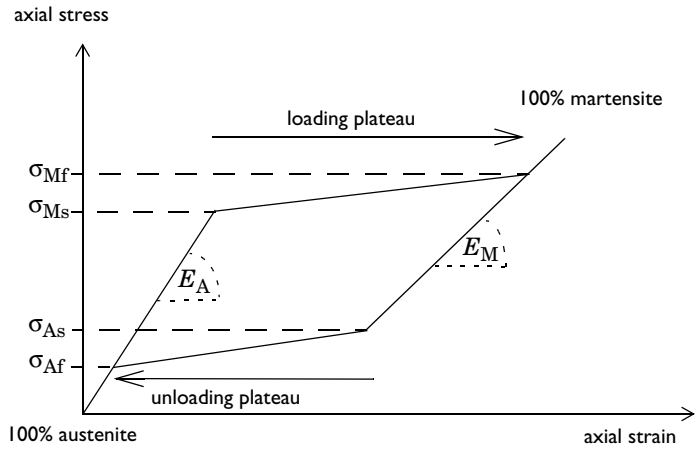


Figure 3-33: Uniaxial stress–strain curve at constant temperature, Lagoudas parameters.

The relations between the start and finish temperatures and stresses are

$$\sigma_{Ms} = C_M(T_\sigma - M_s), \sigma_{Mf} = C_M(T_\sigma - M_f)$$

$$\sigma_{As} = C_A(T_\sigma - A_s), \sigma_{Af} = C_A(T_\sigma - A_f)$$

here, T_σ is the constant temperature at which the stress–strain curve for the pseudoelastic effect was measured. This curve is observed when the temperature T_σ is higher than the austenite finish temperature A_f.

The transition temperatures defined in terms of transition stresses read

$$M_s = T_\sigma - \frac{\sigma_{Ms}}{C_M}, M_f = T_\sigma - \frac{\sigma_{Mf}}{C_M}$$

$$A_s = T_\sigma - \frac{\sigma_{As}}{C_A}, A_f = T_\sigma - \frac{\sigma_{Af}}{C_A}$$

The two slopes of austenite and martensite limit curves C_A and C_M are sometimes referred to as the *Clausius–Clapeyron constants* are measured in a strain–temperature curve at constant stress (Ref. 41).

EQUIVALENT MATERIAL PROPERTIES FOR THE SOUZA–AURICCHIO MODEL

Lagoudas model contains more parameters than the Souza–Auricchio model. The latter model assumes that the slope of limit curves to martensite and austenite are equal, this is, $C_M = C_A$. These two parameters influence the slopes of the loading and unloading plateaus, respectively. For the Souza–Auricchio model, the slope of limit curve β can be taken from the martensite limit curve C_M, or as the average between the austenite and martensite limit curves.

The function $\tau_m(T)$ corresponds to the temperature-dependent martensite to austenite equilibrium stress. It is defined from the slope of the limit curve β and the reference temperature T^*. At the current temperature T_σ it equals

$$\tau_m(T_\sigma) = \beta\langle T_\sigma - T^* \rangle$$

The relations between the start and finish stresses and Souza–Auricchio parameters are derived from the so-called *limit function*, $F = |\sigma_{tr}| - \sigma_0$ evaluated at a constant temperature T_σ

$$\sigma_{Ms} = \sigma_0 + \tau_m(T_\sigma)\,,\ \sigma_{Mf} = \sigma_0 + \tau_m(T_\sigma) + H_k\varepsilon_{tr,max}$$

$$\sigma_{As} = -\sigma_0 + \tau_m(T_\sigma) + H_k\varepsilon_{tr,max}\,,\ \sigma_{Af} = -\sigma_0 + \tau_m(T_\sigma)$$

The elastic radius σ_0 represents half of the height of the hysteresis loop, and it depends on the temperature at which the superelastic effect was measured

$$\sigma_0 = \frac{\sigma_{Mf} - \sigma_{As}}{2} = \frac{\sigma_{Ms} - \sigma_{Af}}{2}$$

The hardening parameter H_k relates the slope in the loading plateau with the difference between the *martensite finish stress*, σ_{Mf}, and the *martensite start stress*, σ_{Ms}

$$\sigma_{Mf} = \sigma_{Ms} + H_k\varepsilon_{tr,max}$$

here, $\varepsilon_{tr,max}$ is the maximum transformation strain measured in a strain–temperature curve at constant stress (Ref. 40).

The Souza–Auricchio model assumes that the width of the transition zones in the phase diagram are equal, that is, the temperature difference $A_f - A_s$ and $M_s - M_f$ are the same, which is not the case in the Lagoudas model. In the Souza–Auricchio model,

the reference temperature T^* is taken as the average temperature between the martensite start temperature and the austenite finish temperature

$$T^* = \frac{M_s + A_f}{2}$$

Considering, $C_M = C_A = \beta$, it is possible to write the reference temperature T^* in terms of the martensite start stress, the austenite finish stress, and the measurement temperature T_σ

$$T^* = T_\sigma - \frac{\tau_m(T_\sigma)}{\beta} = T_\sigma - \frac{\sigma_{Ms} + \sigma_{Af}}{2\beta},$$

Figure 3-34 illustrates the material parameters for the Souza–Auricchio model.

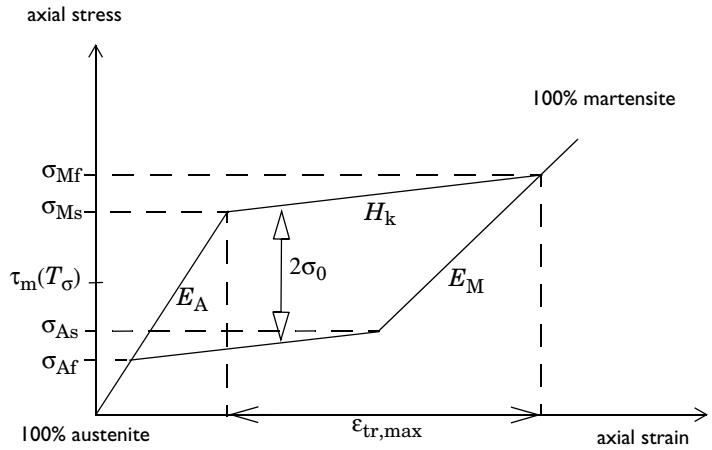


Figure 3-34: Uniaxial stress–strain curve at constant temperature, Souza–Auricchio parameters.

DISSIPATION

Since phase transformation is an inelastic process, the dissipated energy density can be calculated by integrating the pseudo-rate given by

$$\dot{W}_{tr} = \sigma : \dot{\varepsilon}_{tr}$$

For the Souza–Auricchio model the dissipated energy density rate is then defined by

$$\dot{W}_{tr} = \lambda_p \sigma : \frac{\partial F}{\partial \sigma_{tr}}$$

For the Lagoudas model it is defined by

$$\dot{W}_{tr} = \dot{\xi}\sigma:\Lambda$$

As plasticity is rate independent, the transformation dissipation density W_{tr} is obtained after integrating an extra variable in the plastic flow rule.

The total energy dissipated by plasticity in a given volume can be calculated by the volume integration of the plastic dissipation density W_{tr}.

When the **Calculate dissipated energy** checkbox is selected, the transformation dissipation density is available under the variable `solid.Wtr` and the total transformation dissipation under the variable `solid.Wtr_tot`.

Electromechanics

The Electromechanics, Solid Interface theory is described in this section:

- Overview of Forces in Continuum Mechanics
- The Electromagnetic Stress Tensor
- Large Deformations
- About Different Stress Forms
- References for the Electromechanics Interface

OVERVIEW OF FORCES IN CONTINUUM MECHANICS

The deformation of elastic materials subject to external loads is described by the momentum balance equation:

$$\rho\frac{d^2\mathbf{u}}{dt^2} = \nabla\cdot\sigma + \mathbf{f}_v$$

where ρ is the density, $\mathbf{u}$ denotes the structural displacement vector, σ is the stress tensor, and $\mathbf{f}_v$ is an external volume force (such as gravity).

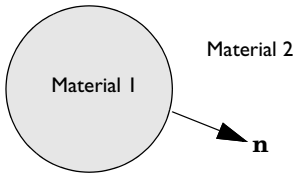
In the stationary case, there is no acceleration, and the equation representing the force balance is

$$\mathbf{0} = \nabla\cdot\sigma + \mathbf{f}_v$$

The stress tensor must be continuous across a stationary boundary between two materials. This corresponds to the equation

$$(\sigma_2 - \sigma_1)\mathbf{n} = \mathbf{0} \tag{3-163}$$

where σ_1 and σ_2 represent the stress tensor in Materials 1 and 2, respectively, and $\mathbf{n}_1$ is the normal vector pointing out from the domain containing Material 1.



THE ELECTROMAGNETIC STRESS TENSOR

Within a vacuum or other medium, forces between charged bodies can be computed on the assumption that a fictitious state of stress exists within the field. Historically the consistency of this approach led 19th-century physicists to postulate the existence of the ether, a ubiquitous medium through which electromagnetic forces propagate. While these ideas have been superseded by the development of special relativity, the use of an electromagnetic stress tensor (also known as the Maxwell stress tensor) remains an accurate and convenient technique to compute electromagnetic forces.

Maxwell stresses exists in all dielectric materials, including air and even free space. However, the force magnitude is rather small, and it usually can cause significant deformations only at small dimension.

The electromagnetic stress tensor in a vacuum (in the absence of magnetic fields) is given by (Ref. 151):

$$\sigma_{\mathrm{EM,\,V}} = \varepsilon_0 \mathbf{E} \otimes \mathbf{E} - \frac{1}{2}(\varepsilon_0 \mathbf{E} \cdot \mathbf{E})I \tag{3-164}$$

Where $\mathbf{E}$ is the electric field, I is the identity tensor, ε_0 is the permittivity of free space, and

$$(\mathbf{E} \otimes \mathbf{E})_{ij} = E_i E_j$$

Within an isotropic linear dielectric under small deformations, the electric displacement vector is related to the electric field as (Ref. 152):

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \varepsilon_0 \chi \mathbf{E}$$

where the electric susceptibility χ can be a function of the mechanical strain in the material

$$\chi = \chi_0(I - 2\mathrm{T} + \mathrm{tr}(\mathrm{T})I)$$

where the small strain tensor is given by

$$\mathrm{T} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$$

The corresponding electromagnetic tress tensor takes the following form:

$$\sigma_{\mathrm{EM}} = \varepsilon_0\varepsilon_r\mathbf{E}\otimes\mathbf{E} - \frac{1}{2}\varepsilon_0\varepsilon_r(\mathbf{E}\cdot\mathbf{E})I \tag{3-165}$$

where the relative permittivity of the material with the absence of deformation is introduced as $\varepsilon_r = 1 + \chi_0$. It can be written equivalently as

$$\sigma_{\mathrm{EM}} = \sigma_{\mathrm{EM,V}} + \varepsilon_0\chi_0\left[\mathbf{E}\otimes\mathbf{E} - \frac{1}{2}(\mathbf{E}\cdot\mathbf{E})I\right]$$

where the two terms represent, respectively, the contributions from: the underlying free space and the material polarization.

The Electromechanics, Solid coupling feature applies the material and underlying free space contributions to the stress as a body load. In weak form: $-\sigma_{\mathrm{EM}}$:test(T).

LARGE DEFORMATIONS

For finite deformations, the expressions for the electromagnetic stress and material polarization can be derived using the following thermodynamic potential called electric enthalpy:

$$H_{\mathrm{eme}} = W_s(C) - \frac{1}{2}[\varepsilon_0\varepsilon_r JC^{-1}]:(\mathbf{E}_{\mathrm{m}}\otimes\mathbf{E}_{\mathrm{m}})$$

where the subscript m indicates that the vector components must be taken on the material frame, and the right Cauchy–Green deformation tensor is

$$C = F^TF$$

with $F = \nabla\mathbf{u} + I$ and $J = \det(F)$. The mechanical energy function $W_s(C)$ depends on the solid model used.

The total second Piola–Kirchhoff stress tensor is given by

$$S = 2\frac{\partial H_{\mathrm{eme}}}{\partial C}$$

and the electric displacement can be calculated as

$$\mathbf{D}_{\mathrm{m}} = -\frac{\partial H_{\mathrm{eme}}}{\partial \mathbf{E}_{\mathrm{m}}}$$

The electromagnetic stress tensor can be used to compute the forces acting on a dielectric body. From Equation 3-163 the balance of forces at the surface of a dielectric body (material 1) in vacuum or air (material 2) implies:

$$(\sigma_2 - \sigma_1)\mathbf{n} = \mathbf{0}$$

where the total stress tensors in media 1 and 2 are, respectively:

$$\sigma_1 = \sigma_{EM} + \sigma_{\mathrm{mech}}$$
$$\sigma_2 = \sigma_{EM,V} - p_2 I$$

where σ_{m} is a mechanical component of the total stress and the ambient pressure p_2 in to surrounding air or other medium (if present). Using Equation 3-174:

$$\sigma_1\mathbf{n} = -p_2\mathbf{n} - \left(\frac{1}{2}\varepsilon_0\mathbf{E}\cdot\mathbf{E}\right)\mathbf{n} + \varepsilon_0(\mathbf{n}\cdot\mathbf{E})\mathbf{E}$$

where the electric field $\mathbf{E}$ is computed in material 2.

The Electromechanics, Solid coupling feature applies the above defined traction as a boundary load on the boundaries which are external for the coupling feature selection but internal for the corresponding Electrostatics interface.

COMSOL Multiphysics does not explicitly include the ambient pressure definition on the Electromechanical Forces coupling feature. However, it is possible to add an additional surface force to the corresponding Solid Mechanics interface if the pressure is known or computed by another physics interface.

ABOUT DIFFERENT STRESS FORMS

There exists a long-lasting controversy in scientific literature about the definition of electromagnetic forces acting on solids. An extensive review can be found in Ref. 143. Many classic textbooks (for example, Ref. 151 and Ref. 152) operate with the so-called Minkowski stress tensor that is usually written as

$$\sigma_{EM} = \mathbf{D} \otimes \mathbf{E} - \frac{1}{2}(\mathbf{D} \cdot \mathbf{E})I$$

The corresponding electromagnetic body force can be written as

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \rho_e \mathbf{E} - \frac{1}{2}(\mathbf{E} \otimes \mathbf{E}) : \nabla \chi$$

which sometimes is referred to as the Korteweg–Helmholtz force. For homogeneous materials without deformation, one has $\nabla \chi = 0$. Hence, in the absence of free charges ($\rho_e = 0$), the body force becomes zero. Thus, the whole electromechanical load on the solid is due to the Maxwell stress jumps at the boundaries between domains with different material properties.

There are two most often used alternatives. The first one is called Einstein–Laub stress tensor:

$$\sigma_{EM} = \mathbf{D} \otimes \mathbf{E} - \frac{1}{2}(\varepsilon_0 \mathbf{E} \cdot \mathbf{E})I$$

This form is widely accepted in modern electroelasticity and material science, see Ref. 144. The corresponding body force can be written as

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \rho_e \mathbf{E} + (\mathbf{P} \cdot \nabla)\mathbf{E}$$

which is also called Kelvin force. Note that it is nonzero as soon as there are electric field variation and polarization within the material:

$$\mathbf{P} = \varepsilon_0 \chi \mathbf{E}$$

The other alternative is Chu stress tensor:

$$\sigma_{EM} = \varepsilon_0 \mathbf{E} \otimes \mathbf{E} - \frac{1}{2}(\varepsilon_0 \mathbf{E} \cdot \mathbf{E})I \tag{3-166}$$

and the corresponding electromagnetic body force is given by

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \rho_e \mathbf{E} - (\nabla \cdot \mathbf{P})\mathbf{E}$$

Note that in contrast to the above equation, the expression for the stress tensor itself (Equation 3-166) seems to ignore the material polarization properties. However, if the frame difference is important because of the deformations, one has

$$\mathbf{D}_\mathrm{m} = \varepsilon_0 J C^{-1}\mathbf{E}_\mathrm{m} + \varepsilon_0 \chi_\mathrm{m}\mathbf{E}_\mathrm{m} \tag{3-167}$$

The electric enthalpy can be written as

$$H_\mathrm{eme} = W_s(C) - \frac{1}{2}\mathbf{D}_\mathrm{m}\cdot\mathbf{E}_\mathrm{m}$$

Then,

$$\mathbf{D}_\mathrm{m} = -\frac{\partial H_\mathrm{eme}}{\partial \mathbf{E}_\mathrm{m}}$$

and the electromechanical second Piola–Kirchhoff stress tensor is given by

$$S_\mathrm{EM} = 2\frac{\partial H_\mathrm{eme}}{\partial C} = JF^{-1}\sigma_\mathrm{EM}F^{-T} \tag{3-168}$$

where σ_EM is given by Equation 3-166, which shows the consistency of the theory. The key assumption behind the derivation of Equation 3-168 is that the material components of the electric susceptibility χ_m are independent of deformations.

COMSOL Multiphysics provides a choice of the electromagnetic stress model, which is available in the Electromechanics, Solid coupling feature. Any of the above presented three options can be selected; the default choice is the Minkowski stress tensor.

REFERENCES FOR THE ELECTROMECHANICS INTERFACE

1. J.A. Stratton, *Electromagnetic Theory*, Cambridge, MA, 1941.

2. L.D. Landau and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Pergamon Press, pp. 69–73, 1960.

3. D.J. Griffiths, "Resource Letter EM-1: Electromagnetic Momentum," *Am. J. Phys.*, vol. 80, pp. 7–18, 2012.

4. L. Dorfmann, R.W. Ogden, *Mechanics and Electrodynamics of Electroelastic and Magnetoelastic Interactions*, Springer, 2014.

Piezoelectricity

The piezoelectric effect manifests itself as a transfer of electric to mechanical energy and vice versa. It is present in many crystalline materials, while some materials such as quartz, Rochelle salt, and lead titanate zirconate ceramics display the phenomenon strongly enough for it to be of practical use.

The *direct piezoelectric effect* consists of an electric polarization in a fixed direction when the piezoelectric crystal is deformed. The polarization is proportional to the deformation and causes an electric potential difference over the crystal.

The *inverse piezoelectric effect*, on the other hand, constitutes the opposite of the direct effect. This means that an applied potential difference induces a deformation of the crystal.

PIEZOELECTRIC CONSTITUTIVE RELATIONS

It is possible to express the relation between the stress, strain, electric field, and electric displacement field in either a *stress–charge form* or *strain–charge form*:

Stress–Charge

$$T = c_{\mathrm{E}}\mathrm{S} - e^{T}\mathbf{E}$$
$$\mathbf{D} = e\mathrm{S} + \varepsilon_0\varepsilon_{\mathrm{rS}}\mathbf{E}$$

Strain–Charge

$$S = s_{\mathrm{E}}\mathrm{T} + d^{T}\mathbf{E}$$
$$\mathbf{D} = d\mathrm{T} + \varepsilon_0\varepsilon_{\mathrm{rT}}\mathbf{E}$$

In the above relations, the naming convention used in piezoelectricity theory is assumed, so that the structural strain is denoted by S, and the stress is denoted by T. Thus, the naming convention differs in piezoelectricity theory compared to structural mechanics theory.

!	The Piezoelectric Material uses the structural mechanics nomenclature. The strain is named ε (instead of S) and the stresses are denoted by either σ or S (instead of T). This makes the names consistent with those used in the other structural mechanics interfaces.

The constitutive relation using COMSOL Multiphysics symbols for the different constitutive forms are thus:

STRESS–CHARGE

$$\begin{aligned}\sigma &= c_{\mathrm{E}}\varepsilon - e^{T}\mathbf{E}\\ \mathbf{D} &= e\varepsilon + \varepsilon_{0,\,\mathrm{vac}}\varepsilon_{\mathrm{rS}}\mathbf{E}\end{aligned} \tag{3-169}$$

The stress–charge form is always used in the variational formulation (weak equation form) which COMSOL Multiphysics uses for discretization and computation.

STRAIN–CHARGE

$$\begin{aligned}\varepsilon &= s_{\mathrm{E}}\sigma + d^{T}\mathbf{E}\\ \mathbf{D} &= d\sigma + \varepsilon_{0,\,\mathrm{vac}}\varepsilon_{\mathrm{rT}}\mathbf{E}\end{aligned} \tag{3-170}$$

Most material data appear in the strain–charge form, and it can easily be transformed into the stress–charge form. In COMSOL Multiphysics both constitutive forms can be used; simply select one, and the software makes any necessary transformations. The following equations transform strain–charge material data to stress–charge data:

$$\begin{aligned}c_{\mathrm{E}} &= s_{\mathrm{E}}^{-1}\\ e &= d\,s_{\mathrm{E}}^{-1}\\ \varepsilon_{\mathrm{rS}} &= \varepsilon_{\mathrm{rT}} - d\,s_{\mathrm{E}}^{-1}\,d^{T}/\varepsilon_{0,\,\mathrm{vac}}\end{aligned} \tag{3-171}$$

All the necessary material data inputs are placed within the Piezoelectric Material feature under the Solid Mechanics interface, which are added automatically when adding a predefined Piezoelectricity multiphysics interface. Such node can be also added manually under any Solid Mechanics interface similar to all other material model features. The piezoelectric material uses the Voigt notation for the anisotropic material data, as customary in this field. More details about the data ordering can be found in Orthotropic and Anisotropic Materials section.

GOVERNING EQUATIONS

The equations of Piezoelectricity combine the momentum equation Equation 3-189 with the charge conservation equation of Electrostatics,

$$\nabla\cdot\mathbf{D} = \rho_{\mathrm{V}} \tag{3-172}$$

where the ρ_{V} is the electric charge concentration. The electric field is computed from the electric potential V as

$$\mathbf{E} = -\nabla V$$

In both Equation 3-172 and Equation 3-189, the constitutive relations Equation 3-169 are used, which makes the resulting system of equations closed. The dependent variables are the structural displacement vector $\mathbf{u}$ and the electric potential V.

WAVE PROPAGATION IN PIEZOELECTRIC MEDIA

In case of geometric linearity, the governing equations for a linear piezoelectric medium of any anisotropy can be written in terms of the structural displacement vector $\mathbf{u}$ and electric potential V as

$$\begin{aligned}\rho\frac{\partial^2\mathbf{u}}{\partial t^2} &= \nabla\cdot(\mathsf{C}:\nabla\mathbf{u}+\nabla V\cdot\mathsf{e})\\ \nabla\cdot(\mathsf{e}:\nabla\mathbf{u}-\varepsilon_0\varepsilon_r\cdot\nabla V) &= 0\end{aligned} \tag{3-173}$$

The following symmetries hold for the constitutive tensor coefficients:

$$\mathsf{C}^{ijkl} = \mathsf{C}^{jikl} = \mathsf{C}^{klij}, \quad \mathsf{e}^{ikl} = \mathsf{e}^{ilk}, \quad \varepsilon_r^{kl} = \varepsilon_r^{lk}.$$

Piezoelectric material data inputs in COMSOL Multiphysics are based on the matrix representation of higher-order tensors using Voigt notations. Thus, to obtain coefficients of 6-by-3 coupling matrix e^{im}, the first index of the coupling tensor components e^{ikl} remains the same, while the last two indices should be replaced by a single index obtained using special rules. Similarly, for the stiffness tensor, the first pair and the second pair of indices must be replaced by a single index used instead of each pair. See Orthotropic and Anisotropic Materials for the index transformation rules.

The constitutive relations in tensor notations can be derived using a thermodynamic potential called the electric enthalpy:

$$H_e(\nabla\mathbf{u},\nabla V) = \frac{1}{2}\nabla\mathbf{u}:\mathsf{C}:\nabla\mathbf{u}+\nabla V\cdot\mathsf{e}:\nabla\mathbf{u}-\frac{1}{2}\varepsilon_0\nabla V\cdot\varepsilon_r\cdot\nabla V$$

Thus,

$$\sigma = \frac{\partial H_e}{\partial \nabla \mathbf{u}} = \mathsf{C} : \nabla \mathbf{u} + \nabla V \cdot \mathsf{e}$$

$$\mathbf{D} = \frac{\partial H_e}{\partial \nabla V} = \mathsf{e} : \nabla \mathbf{u} - \varepsilon_0 \varepsilon_r \cdot \nabla V$$

Since the relations in Equation 3-173 are linear, they possess the following time-harmonic wave solutions:

$$\mathbf{u} = \mathbf{u}_n e^{i\omega t - i\mathbf{k}\cdot\mathbf{r}}$$

$$V = \mathbf{u}_n e^{i\omega t - i\mathbf{k}\cdot\mathbf{r}}$$

where $\mathbf{k} = k\mathbf{n}$ is the wave number vector, $\mathbf{n}$ is the direction vector that defines the wavefront propagation direction. The wavefront is an imaginary line connecting solid particles of the same phase. The velocity of such wavefront in the direction normal to it is given by the phase velocity $c = \omega/k$.

A generalization of Christoffel's equation (Equation 3-28) can then be done:

$$\rho c^2 \mathbf{u}_n = \Gamma_n \cdot \mathbf{u}_n$$

$$\mathbf{u}_n = \mathbf{e}_n \cdot \mathbf{u}_n / \varepsilon_n$$

where the generalized Christoffel's tensor is defined as

$$\Gamma_n = \mathbf{n} \cdot \mathsf{C} \cdot \mathbf{n} + \mathbf{e}_n \otimes \mathbf{e}_n / \varepsilon_n$$

and

$$\mathbf{e}_n = \mathbf{n} \cdot \mathsf{e} \cdot \mathbf{n}$$

$$\varepsilon_n = \varepsilon_0 \mathbf{n} \cdot \varepsilon_r \cdot \mathbf{n}$$

The generalized Christoffel's equation can be considered as an eigenvalue problem. Thus, to have a nontrivial solution, the phase velocity must satisfy:

$$\det(\Gamma_n/\rho - c^2 I) = 0$$

which is called the dispersion relation. This is in general case a cubic polynomial in terms of c^2, which has three roots $c^2 = c_j^2(\Gamma_n/\rho)$. Thus, for an arbitrary anisotropic

medium, three waves with different phase velocities can propagate in each given direction.

However, most piezoelectric materials exhibit only moderate degree of anisotropy. The material data is usually given in a coordinate system, in which it is ether orthotropic or even transversely isotropic so that only one direction (poling direction) differs from the two others. In this case, the expressions for the phase velocity can be computed analytically for waves of certain types, polarizations, and directions of propagations.

For example, the pressure wave propagating in the X direction is a particular solution, for which

$$\begin{aligned} u &= \hat{u}\exp[-ik_X(X-c_Xt)] \\ v &= 0 \\ w &= 0 \\ V &= \hat{V}\exp[-ik_X(X-c_Xt)] \end{aligned}$$

The corresponding phase velocity is given by

$$c_{\mathrm{p}X} = \frac{1}{\rho}\left(c_{E,11} + \frac{e_{X1}^2}{\varepsilon_0\varepsilon_{\mathrm{r}S,XX}}\right)^{1/2}$$

Here, the material data coefficients are given in the same matrix form notation that is used in COMSOL Multiphysics for data input.

The shear wave propagation in the X direction and with XY-plane polarization is a solution such that

$$\begin{aligned} u &= 0 \\ v &= \hat{v}\exp[-ik_X(X-c_Xt)] \\ w &= 0 \\ V &= \hat{V}\exp[-ik_X(X-c_Xt)] \end{aligned}$$

and the corresponding phase velocity is computed as

$$c_{\mathrm{s}XY} = \frac{1}{\rho}\left(c_{E,66} + \frac{e_{X6}^2}{\varepsilon_0\varepsilon_{\mathrm{r}S,XX}}\right)^{1/2}$$

If the wave propagation is initiated by a small perturbation that is initially localized in space, the solution can be found using Fourier and Laplace transforms, and it will

represent a so-called wave packet. The wave packet will propagate with the group velocity given by

$$\frac{\partial \omega_j}{\partial \mathbf{k}} = \frac{1}{2\rho c_j}\frac{\partial}{\partial \mathbf{n}}(\mathbf{u}_{\mathrm{n},j} \cdot \Gamma_\mathrm{n} \cdot \mathbf{u}_{\mathrm{n},j}), \quad j = 1, 2, 3$$

where $\mathbf{u}_{\mathrm{n},j}$ is the wave polarization vector that is the eigenvector corresponding to the eigenvalue solution $c_j^2(\Gamma_\mathrm{n}/\rho)$ of the Christoffel's equation.

COMSOL Multiphysics provides predefined variables for the phase and group velocities for waves of different types propagating in any chosen direction. These variables do not affect the solution as such, but are available during result presentation if the Wave Speeds node has been added to the material.

The wave speed variables can be found in the **Wave speeds** folder under **Solid Mechanics** in the **Replace Expression** tree.

PIEZOELECTRIC DISSIPATION

In order to define dissipation in the piezoelectric material for a time-harmonic analysis, all material properties in the constitutive relations can be complex-valued matrices, where the imaginary part defines the dissipative function of the material.

Complex-valued data can be defined directly in the fields for the material properties, or a real-valued material X and a set of loss factors η_X can be defined, which together form the complex-valued material data

$$\tilde{X} = X(1 \pm j\eta_X)$$

The sign in the complex damping terms are defined as

$$\tilde{c}_\mathrm{E} = (1 + i\eta_{\mathrm{cE}})c_\mathrm{E}$$
$$\tilde{s}_\mathrm{E} = (1 - i\eta_{\mathrm{sE}})s_\mathbf{E}$$
$$\tilde{\varepsilon}_{\mathrm{rS}} = (1 - i\eta_{\varepsilon\mathrm{S}})\varepsilon_{\mathbf{rS}}$$
$$\tilde{\varepsilon}_{\mathrm{rT}} = (1 - i\eta_{\varepsilon\mathrm{T}})\varepsilon_{\mathbf{rT}}$$
$$\tilde{e} = (1 + i\eta_\mathbf{e})e$$
$$\tilde{d} = (1 + i\eta_\mathbf{d})d$$

In the four first expressions, the choice of sign is a result of the single physics observations that

- Strain lags behind stress.
- Electric displacement (or polarization) lags behind the electric field.

For these cases, it is thus necessary that η_X is positive in order for the material to be thermodynamically stable with the chosen signs.

The sign of the coupling losses requires more considerations, and the chosen sign must be considered as a definition. All values of η_e and η_d does not necessarily have to be positive. For some simple theoretical cases with isotropic loss factors, it can however be shown that the definition above is reasonable:

- Consider a material which in strain–charge form only has mechanical damping. Thus, d is real, and the second transformation law in Equation 3-171 shows that e must have a positive loss factor.
- Consider a material with only coupling loss. Then the second transformation law in Equation 3-171 shows that d and e must have the same loss factor multiplier. From the third transformation law it can be inferred that η_e and η_d in this case must be positive in order for the material to have an appropriate phase lag between **D** and **E**.

In a real material, all types of losses occur simultaneously. Then, the criteria for allowable values of the loss factors become complicated, particularly if they are not isotropic.

It is also possible to define the electric conductivity of the piezoelectric material, σ. Electric conductivity appears as an additional term in the variational formulation (weak equation form). The conductivity does not change during transformation between the formulations.

The energy dissipation modeling is also available in time domain. The options are: dielectric dispersion for the electrical part, and Rayleigh damping for the mechanical and coupling parts of the problem. The total dissipated energy can be computed as a function of time.

	Piezoelectric Losses

INITIAL STRESS, STRAIN, AND ELECTRIC DISPLACEMENT

Using the functionality available under the Piezoelectric Material feature and Solid Mechanics interface, one can define initial stress (S_0), initial strain (ε_0), and remanent electric displacement ($\mathbf{D}_r$) for models. In the constitutive relation for piezoelectric material these additions appear in the stress–charge formulation:

$$\sigma = c_E(\varepsilon - \varepsilon_0) - e^T\mathbf{E} + \sigma_0$$
$$\mathbf{D} = e(\varepsilon - \varepsilon_0) + \varepsilon_{0,vac}\varepsilon_{rS}\mathbf{E} + \mathbf{D}_r$$

MULTIPLICATIVE FORMULATION FOR PIEZOELECTRICITY

The total deformation gradient is computed from the structural displacement field as

$$F = \nabla\mathbf{u} + I$$

and the right Cauchy–Green is defined as $C = F^TF$.

The decomposition between elastic and inelastic deformation is made using a multiplicative decomposition of the deformation gradient

$$F_{el} = FF_{in}^{-1}$$

where the inelastic deformation tensor depends on the inelastic process, such as piezoelectric effect and thermal expansion $F_{in} = F_{in}(\mathbf{E}, T)$.

The elastic right Cauchy–Green deformation tensor is then computed from F_{el}

$$C_{el} = F_{el}^TF_{el} = F_{in}^{-T}CF_{in}^{-1}$$

and the elastic Green–Lagrange strain tensor is computed as

$$\varepsilon_{el} = \frac{1}{2}(C_{el} - I)$$

Note that $C_{el} = C_{el}(C, \mathbf{E}, T)$.

The strain tensor C, the electric field $\mathbf{E}$, and the temperature T are used as the state variables. The free energy density in the undeformed configuration is defined as

$$W(C, \mathbf{E}, T) = J_{in}W_{el}(C_{el}) + W_{in}(\mathbf{E}, T) + JW_{vac}(\mathbf{E}_s)$$

where $J = \det(F)$ and $J_{in} = \det(F_{in})$.

For structurally linear material,

$$W_{\text{el}} = \frac{1}{2}\varepsilon_{\text{el}} : \mathfrak{c} : \varepsilon_{\text{el}}$$

where $\mathfrak{c}$ is the elasticity tensor.

The free space contribution is computed as

$$W_{\text{vac}}(\mathbf{E}_{\text{s}}) = -\frac{1}{2}\varepsilon_{0,\,\text{vac}}(\mathbf{E}_{\text{s}} \cdot \mathbf{E}_{\text{s}}) = -\frac{1}{2}\varepsilon_{0,\,\text{vac}} C^{-1}(\mathbf{E} \cdot \mathbf{E})$$

where $\mathbf{E}_{\text{s}}$ and $\mathbf{E}$ are the electric fields in the deformed (spatial) and undeformed configuration, respectively.

The second Piola–Kirchhoff stress is computed as

$$S = 2\frac{\partial W}{\partial C} = 2\frac{\partial}{\partial C}(J_{\text{in}} W_{\text{el}}) + S_{\text{M}}$$

The Maxwell stress S_{M} is related to the polarization of the free space occupied by the deformed body, and it is computed as

$$S_{\text{M}} = 2\frac{\partial}{\partial C}(J W_{\text{vac}}) = \varepsilon_{0,\,\text{vac}} J\left(C^{-1}(\mathbf{E} \otimes \mathbf{E}) - \frac{1}{2}((C^{-1}\mathbf{E}) \cdot \mathbf{E})I\right)C^{-1}$$

The inelastic deformation gradient is further decomposed into a thermal and piezoelectric parts as

$$F_{\text{in}} = F_{\text{th}}(T) F_{\text{pze}}(\mathbf{E})$$

The piezoelectric strain tensor is introduced as $\varepsilon_{\text{pze}} = \mathbf{E} \cdot \mathfrak{d}$, where $\mathfrak{d}$ is the piezoelectric coupling tensor. The piezoelectric deformation gradient is modeled as

$$F_{\text{pze}} = I + \varepsilon_{\text{pze}} + O(\|\varepsilon_{\text{pze}}\|^2)$$

Introduce

$$C_{\text{tel}} = F_{\text{th}}^{-T} C F_{\text{th}}^{-1}$$

$$\varepsilon_{\text{tel}} = \frac{1}{2}(C_{\text{tel}} - I)$$

$$S_{\text{tel}} = \mathfrak{c} : \varepsilon_{\text{tel}}$$

which are the strain and stress tensors in the intermediate configuration, where the thermal expansion part have been removed. These quantities are independent of the electric field $\mathbf{E}$.

Then,

$$\varepsilon_{\mathrm{el}} = \varepsilon_{\mathrm{tel}} - C_{\mathrm{tel}}\varepsilon_{\mathrm{pze}} + O(\|\varepsilon_{\mathrm{pze}}\|^2)$$

and

$$J_{\mathrm{in}}W_{\mathrm{cl}} = \frac{1}{2}J_{\mathrm{th}}\varepsilon_{\mathrm{tel}} : S_{\mathrm{tel}} + J_{\mathrm{th}}W_{\mathrm{pze}} + O(\|\varepsilon_{\mathrm{pze}}\|^2)$$

where

$$W_{\mathrm{pze}} = \left(\frac{1}{2}\mathrm{tr}(\varepsilon_{\mathrm{pze}})\varepsilon_{\mathrm{tel}} - C_{\mathrm{tel}}\varepsilon_{\mathrm{pze}}\right) : S_{\mathrm{tel}}$$

The stress is computed then as

$$S = J_{\mathrm{th}}F_{\mathrm{th}}^{-1}\left(S_{\mathrm{tel}} + \frac{\partial W_{\mathrm{pze}}}{\partial \varepsilon_{\mathrm{tel}}}\right)F_{\mathrm{th}}^{-T} + S_{\mathrm{M}} + O(\|\varepsilon_{\mathrm{pze}}\|^2)$$

The electric displacement (in the undeformed configuration) is computed as

$$\mathbf{D} = -\frac{\partial W}{\partial \mathbf{E}} = \varepsilon_{0,\,\mathrm{vac}}JC^{-1}\mathbf{E} + \mathbf{P} + J_{\mathrm{th}}\frac{\partial W_{\mathrm{pze}}}{\partial \mathbf{E}} + O(\|\varepsilon_{\mathrm{pze}}\|^2)$$

where the material polarization is given by

$$\mathbf{P} = -\frac{\partial W_{\mathrm{in}}}{\partial \mathbf{E}}$$

For electrically and thermally linear material, it is computed as

$$\mathbf{P} = \varepsilon_{0,\,\mathrm{vac}}(\varepsilon_{\mathrm{r}} - I)\mathbf{E} + \mathbf{p}\theta$$

where ε_{r} is the relative electric permittivity, $\mathbf{p}$ is the pyroelectric coefficient, and $\theta = T - T_{\mathrm{ref}}$ is the temperature variation.

Magnetomechanics

The theory for The Magnetomechanics, Solid Interface, The Magnetomechanics, No Currents, Solid Interface, The Magnetic–Elastic Interaction in Rotating Machinery

Interface and The Magnetic–Rigid-Body Interaction in Rotating Machinery Interface is given in this section.

THE ELECTROMAGNETIC STRESS TENSOR

Within a vacuum or other medium, forces between magnetized bodies can be computed on the assumption that a fictitious state of stress exists within the field. Historically, the consistency of this approach led 19th-century physicists to postulate the existence of the ether, a ubiquitous medium through which electromagnetic forces propagate. While these ideas have been superseded by the development of particle physics, the use of an electromagnetic stress tensor (also known as the Maxwell stress tensor) remains an accurate and convenient technique to compute electromagnetic forces.

Maxwell stresses exist in all materials, including air and even free space. However, the force magnitude is rather small, and usually it can only cause significant deformations at small dimension.

The electromagnetic stress tensor in a vacuum (in the absence of electric fields) is given by (Ref. 151):

$$\sigma_{\mathrm{EM,\,V}} = \mu_0 \mathbf{H} \otimes \mathbf{H} - \frac{1}{2}\mu_0 (\mathbf{H} \cdot \mathbf{H})\mathrm{I} \tag{3-174}$$

Where $\mathbf{H}$ is the magnetic field, I is the identity tensor, μ_0 is the magnetic permeability of free space, and $\otimes$ is the outer product of two vectors

$$(\mathbf{H} \otimes \mathbf{H})_{ij} = H_i H_j$$

Within an isotropic linear magnetic solid under small deformations, the magnetic flux density vector is related to the magnetic field as (Ref. 152):

$$\mathbf{B} = \mu_0 \mathbf{H} + \mu_0 \chi \mathbf{H}$$

where the magnetic susceptibility χ can be a function of the mechanical strain in the material

$$\chi = \chi_0 (\mathrm{I} - 2\mathrm{T} + \mathrm{tr}(\mathrm{T})\mathrm{I})$$

where χ_0 is the magnetic susceptibility of the material without deformation, and T is the small-strain tensor given by

$$\mathrm{T} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

The corresponding electromagnetic tress tensor takes the following form:

$$\sigma_{EM} = \mu_0\mu_r\mathbf{H}\otimes\mathbf{H} - \frac{1}{2}\mu_0\mu_r(\mathbf{H}\cdot\mathbf{H})\mathrm{I} \tag{3-175}$$

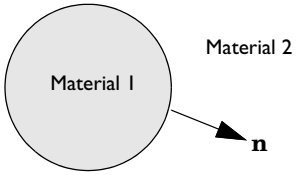
where the relative magnetic permeability is introduced as $\mu = 1+\chi$. It can be written equivalently as

$$\sigma_{EM} = \sigma_{EM,V} + \mu_0\chi\left[\mathbf{H}\otimes\mathbf{H} - \frac{1}{2}(\mathbf{H}\cdot\mathbf{H})\mathrm{I}\right]$$

where the two terms represent the contributions from the underlying free space and the material magnetization, respectively.

The Magnetomechanics, Solid and Magnetic Forces, Rotating Machinery coupling features apply the material and underlying free space contributions to the stress as body load. In weak form: $-\sigma_{EM}$:test(T).

The electromagnetic stress tensor can be used to compute the forces acting on a solid body.



The balance of forces at the surface of the solid (material 1) in vacuum or air (material 2) implies:

$$(\sigma_2 - \sigma_1)\mathbf{n} = \mathbf{0}$$

where the total stress tensors in media 1 and 2 are, respectively:

$$\sigma_1 = \sigma_{EM} + \sigma_{mech}$$
$$\sigma_2 = \sigma_{EM,V} - p_2\mathrm{I}$$

where σ_{mech} is a mechanical component of the total stress in the material, and p_2 is the pressure in the surrounding air (or other medium). Using Equation 3-174:

$$\sigma_1\mathbf{n} = -p_2\mathbf{n} - \frac{1}{2}\mu_0(\mathbf{H}\cdot\mathbf{H})\mathbf{n} + \mu_0(\mathbf{n}\cdot\mathbf{H})\mathbf{H}$$

where the magnetic field $\mathbf{H}$ is computed in material 2.

The Magnetomechanics, Solid; Magnetomechanics, Boundary; and Magnetic Forces, Rotating Machinery coupling features apply the above defined traction as a boundary load on the boundaries which are external for the coupling feature selection but internal for the corresponding magnetic fields interface.

COMSOL Multiphysics does not explicitly include the ambient pressure definition on the coupling features. However, it is possible to add an additional surface force to the corresponding Solid Mechanics interface if the pressure is known or computed by another physics interface.

LARGE DEFORMATIONS

For finite deformations, the expressions for the electromagnetic stress and material magnetization can be derived using the following thermodynamic potential called magnetic enthalpy:

$$W_{\mathrm{EM}} = W_{\mathrm{s}}(\mathrm{C}) - \frac{1}{2}[\mu_0\mu_{\mathrm{r}}J\mathrm{C}^{-1}]:(\mathbf{H}_{\mathrm{m}} \otimes \mathbf{H}_{\mathrm{m}})$$

where the subscript m indicates that the vector components must be taken on the material frame, and the right Cauchy–Green deformation tensor is

$$\mathrm{C} = F^T F$$

with

$$F = \nabla \mathbf{u} + \mathrm{I}$$

and $J = \det(F)$. The mechanical energy function $W_{\mathrm{s}}(\mathrm{C})$ depends on the solid model used.

The total second Piola–Kirchhoff stress tensor is given by

$$\mathrm{S} = 2\frac{\partial W_{\mathrm{EM}}}{\partial \mathrm{C}}$$

and the magnetic flux density vector can be calculated as

$$\mathbf{B}_{\mathrm{m}} = -\frac{\partial W_{\mathrm{EM}}}{\partial \mathbf{H}_{\mathrm{m}}}$$

ABOUT DIFFERENT STRESS FORMS

There exists a long-lasting controversy in scientific literature about the definition of electromagnetic forces acting on solids. An extensive review can be found in Ref. 143. Many classic textbooks (for example, Ref. 151 and Ref. 152) operate with the so-called Minkowski magnetic stress tensor that is usually written as

$$\sigma_{EM} = \mathbf{B} \otimes \mathbf{H} - \frac{1}{2}(\mathbf{B} \cdot \mathbf{H})\mathrm{I}$$

The corresponding electromagnetic body force can be written as

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \mathbf{J} \times \mathbf{B} - \frac{1}{2}(\mathbf{H} \otimes \mathbf{H}) : \nabla\chi$$

which sometimes is referred to as the Korteweg–Helmholtz magnetic force. For homogeneous materials without deformation, one has $\nabla\chi = 0$. Hence, in the absence of electric currents ($\mathbf{J} = 0$), the body force becomes zero. Thus, the whole magnetomechanical load on the solid is due to the Maxwell stress jump at the boundaries between domains with different material properties.

There exists two most often used alternatives. The first one is called the Einstein–Laub stress tensor:

$$\sigma_{EM} = \mathbf{B} \otimes \mathbf{H} - \frac{1}{2}\mu_0(\mathbf{H} \cdot \mathbf{H})\mathrm{I}$$

This form is widely accepted in modern magnetoelasticity and material science, see Ref. 144. The corresponding body force can be written as

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \mathbf{J} \times \mathbf{B} + \mu_0(\mathbf{M} \cdot \nabla)\mathbf{H}$$

which is also called the Kelvin magnetic force. Note that it is nonzero as soon as there are magnetic field variation and magnetization within the material

$$\mathbf{M} = \chi\mathbf{H}$$

The other alternative is the Chu stress tensor:

$$\sigma_{EM} = \mu_0\mathbf{H} \otimes \mathbf{H} - \frac{1}{2}\mu_0(\mathbf{H} \cdot \mathbf{H})\mathrm{I}$$

and the corresponding magnetic body force is given by

$$\mathbf{f} = \nabla \cdot \sigma_{EM} = \mathbf{J} \times \mathbf{B} - \mu_0(\nabla \cdot \mathbf{M})\mathbf{H}$$

so that variations of both the magnetic susceptibility and magnetic field can contribute.

Magnetomechanics, Boundary and Magnetic Forces, Rotating Machinery coupling features always operate with Minkowski stress tensor form.

For Magnetomechanics, Solid coupling feature, the choice of stress tensor is available only when one of the coupled interfaces is Magnetic Fields, No Currents.

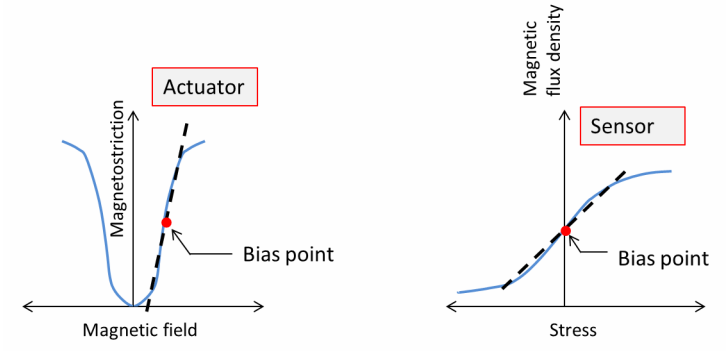
Magnetostriction and Piezomagnetism

Magnetostriction describes the change in dimensions of a material due to a change in its magnetization. This phenomenon is a manifestation of magnetoelastic coupling, which is exhibited by all magnetic materials to some extent. The effects related to magnetoelastic coupling are described by various names. The *Joule effect* describes the change in length due to a change in the magnetization state of the material. This *magnetostrictive effect* is used in transducers for applications in sonars, acoustic devices, active vibration control, position control, and fuel injection systems.

The inverse effect accounts for the change in magnetization due to mechanical stress in the material. This effect is also known *Villari effect*. This effect is mostly useful in sensors.

Magnetostriction has a quantum-mechanical origin. The magneto-mechanical coupling takes place at the atomic level due to spin-orbit coupling. From a system level, the material can be assumed to consist of a number of tiny ellipsoidal magnets which rotate due to the torque produced by the externally applied magnetic field. The

rotation of these elemental magnets produces a dimensional change leading to free strain in the material.



PIEZOMAGNETISM

The magnetostriction has a nonlinear dependence on the magnetic field and the mechanical stress in the material. However, the effect can be modeled using linear coupled constitutive equations if the response of the material consists of small deviations around an operating point (bias point). This type of coupling is reffed to as Piezomagnetic Effect.

It is possible to express the relation between the stress $\mathbf{S}$, strain ε, magnetic field $\mathbf{H}$, and magnetic flux density $\mathbf{B}$ in either a *stress–magnetization form* or *strain–magnetization form*:

Stress–Magnetization

$$\begin{aligned} \mathbf{S} &= c_{\mathrm{H}}\varepsilon - e_{\mathrm{HS}}^{T}\mathbf{H} \\ \mathbf{B} &= e_{\mathrm{HS}}\varepsilon + \mu_0\mu_{\mathrm{rS}}\mathbf{H} \end{aligned}$$

Strain–Magnetization

$$\begin{aligned} \varepsilon &= s_{\mathrm{H}}\mathbf{S} + d_{\mathrm{HT}}^{T}\mathbf{H} \\ \mathbf{B} &= d_{\mathrm{HT}}\mathbf{S} + \mu_0\mu_{\mathrm{rT}}\mathbf{H} \end{aligned}$$

where μ_0 is the magnetic permeability of free space; c_{H} and s_{H} are the stiffness and compliance matrices measured at constant magnetic field, respectively; and μ_{rS} and μ_{rT} are the relative magnetic permeabilities measured at constant strain and constant stress, respectively. The matrices d_{HT} and e_{HS} are called piezomagnetic coupling matrices.

In COMSOL Multiphysics, both constitutive forms can be used; simply select one, and the software makes all necessary transformations. The following equations transform strain–magnetization material data to stress–magnetization data:

$$c_{\mathrm{H}} = s_{\mathrm{H}}^{-1}$$

$$e_{\mathrm{HS}} = d_{\mathrm{HT}} s_{\mathrm{H}}^{-1}$$

$$\mu_{\mathrm{rS}} = \mu_{\mathrm{rT}} - \frac{1}{\mu_0} d_{\mathrm{HT}} s_E^{-1} d_{\mathrm{HT}}^T$$

One can rewrite the system of constitutive relations in the following equivalent form:

$$\mathrm{S} = c_{\mathrm{H}}(\varepsilon - \varepsilon_{\mathrm{me}})$$

$$\mathbf{B} = \mu_0 \mathbf{H} + \mu_0 (\mathbf{M} + \mu_0^{-1} d_{\mathrm{HT}} \mathrm{S})$$

where the magnetostrictive strain is introduced as

$$\varepsilon_{\mathrm{me}} = d_{\mathrm{HT}}^T \mathbf{H}$$

and the material magnetization due to the applied field is given by

$$\mathbf{M} = (\mu_{\mathrm{rT}} - I)\mathbf{H}$$

Hence, the stress in material is proportional to the elastic strain

$$\varepsilon_{\mathrm{el}} = \varepsilon - \varepsilon_{\mathrm{me}}$$

and it contributes to the material magnetization.

All the necessary material data inputs are placed within the Piezomagnetic Material node under the Solid Mechanics interface, which are added automatically when adding a predefined Piezomagnetism multiphysics interface. Such a node can be also added manually under any Solid Mechanics interface similar to all other material model features. The Piezomagnetic Material uses Voigt notation for the anisotropic material data. More details about the data ordering can be found in the Orthotropic and Anisotropic Materials section.

For a crystalline material with tetragonal symmetry, the strain–magnetization form of the constitutive relations is the following:

$$\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{23}\\ 2\varepsilon_{13}\\ 2\varepsilon_{12}\end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & 0 & 0 & 0 \\ s_{12} & s_{11} & s_{13} & 0 & 0 & 0 \\ s_{13} & s_{13} & s_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(s_{11}-s_{12}) \end{bmatrix} \begin{bmatrix} S_{11}\\ S_{22}\\ S_{33}\\ S_{23}\\ S_{13}\\ S_{12}\end{bmatrix} + \begin{bmatrix} 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & d_{33} \\ 0 & d_{15} & 0 \\ d_{15} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} H_1\\ H_2\\ H_3\end{bmatrix}$$

$$\begin{bmatrix} B_1\\ B_2\\ B_3\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{15} & 0 & 0 \\ -\frac{1}{2}d_{33} & -\frac{1}{2}d_{33} & d_{33} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{11}\\ S_{22}\\ S_{33}\\ S_{23}\\ S_{13}\\ S_{12}\end{bmatrix} + \mu_0 \begin{bmatrix} \mu_{11} & 0 & 0 \\ 0 & \mu_{11} & 0 \\ 0 & 0 & \mu_{33} \end{bmatrix} \begin{bmatrix} H_1\\ H_2\\ H_3\end{bmatrix}$$

The following material data corresponds to Terfenol-D at 100 kA/m bias and 30 MPa prestress (Ref. 150):

$$s_{\mathrm{H}} = 10^{-11} \cdot \begin{bmatrix} 4.4 & -1.1 & -1.65 & 0 & 0 & 0 \\ -1.1 & 4.4 & -1.65 & 0 & 0 & 0 \\ -1.65 & -1.65 & 3.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 \end{bmatrix} [1/\mathrm{Pa}]$$

$$d_{\mathrm{HT}} = 10^{-9} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 16.5 & 0 \\ 0 & 0 & 0 & 16.5 & 0 & 0 \\ -4.3 & -4.3 & 8.6 & 0 & 0 & 0 \end{bmatrix} [\mathrm{m/A}] \qquad \mu_{\mathrm{rT}} = \begin{bmatrix} 8.1 & 0 & 0 \\ 0 & 8.1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

MULTIPLICATIVE FORMULATION FOR PIEZOMAGNETISM

The total deformation gradient is computed from the structural displacement field as

$$F = \nabla \mathbf{u} + I$$

and the right Cauchy–Green is defined as $C = F^T F$.

The decomposition between elastic and inelastic deformation is made using a multiplicative decomposition of the deformation gradient

$$F_{\text{el}} = FF_{\text{in}}^{-1}$$

where the inelastic deformation tensor depends on the inelastic process, such as piezomagnetic effect and thermal expansion $F_{\text{in}} = F_{\text{in}}(\mathbf{H}, T)$.

The elastic right Cauchy–Green deformation tensor is then computed from F_{el}

$$C_{\text{el}} = F_{\text{el}}^{T}F_{\text{el}} = F_{\text{in}}^{-T}CF_{\text{in}}^{-1}$$

and the elastic Green–Lagrange strain tensor is computed as

$$\varepsilon_{\text{el}} = \frac{1}{2}(C_{\text{el}} - I)$$

Note that $C_{\text{el}} = C_{\text{el}}(C, \mathbf{H}, T)$.

The strain tensor C, the magnetic field $\mathbf{H}$, and the temperature T are used as the state variables. The free energy density in the undeformed configuration is defined as

$$W(C, \mathbf{H}, T) = J_{\text{in}}W_{\text{el}}(C_{\text{el}}) + W_{\text{in}}(\mathbf{H}, T) + JW_{\text{vac}}(\mathbf{H}_{\text{s}})$$

where $J = \det(F)$ and $J_{\text{in}} = \det(F_{\text{in}})$.

For structurally linear material,

$$W_{\text{el}} = \frac{1}{2}\varepsilon_{\text{el}} : \mathfrak{c} : \varepsilon_{\text{el}}$$

where $\mathfrak{c}$ is the elasticity tensor.

The free space contribution is computed as

$$W_{\text{vac}}(\mathbf{H}_{\text{s}}) = -\frac{1}{2}\mu_{0,\,\text{vac}}(\mathbf{H}_{\text{s}} \cdot \mathbf{H}_{\text{s}}) = -\frac{1}{2}\mu_{0,\,\text{vac}}C^{-1}(\mathbf{H} \cdot \mathbf{H})$$

where $\mathbf{H}_{\text{s}}$ and $\mathbf{H}$ are the magnetic fields in the deformed (spatial) and undeformed configuration, respectively.

The second Piola–Kirchhoff stress is computed as

$$S = 2\frac{\partial W}{\partial C} = 2\frac{\partial}{\partial C}(J_{\text{in}}W_{\text{el}}) + S_{\text{M}}$$

The Maxwell stress S_{M} is related to the magnetization of the free space occupied by the deformed body, and it is computed as

$$S_{\mathrm{M}} = 2\frac{\partial}{\partial C}(JW_{\mathrm{vac}}) = \mu_{0,\,\mathrm{vac}} J\left(C^{-1}(\mathbf{H}\otimes\mathbf{H}) - \frac{1}{2}((C^{-1}\mathbf{H})\cdot\mathbf{H})I\right)C^{-1}$$

The inelastic deformation gradient is further decomposed into a thermal and piezomagnetic parts as

$$F_{\mathrm{in}} = F_{\mathrm{th}}(T)F_{\mathrm{pzm}}(\mathbf{H})$$

The piezomagnetic strain tensor is introduced as $\varepsilon_{\mathrm{pzm}} = \mathbf{H}\cdot\mathrm{d}$, where d is the piezomagnetic coupling tensor. The piezomagnetic deformation gradient is modeled as

$$F_{\mathrm{pzm}} = I + \varepsilon_{\mathrm{pzm}} + O(\|\varepsilon_{\mathrm{pzm}}\|^2)$$

Introduce

$$C_{\mathrm{tel}} = F_{\mathrm{th}}^{-T} C F_{\mathrm{th}}^{-1}$$

$$\varepsilon_{\mathrm{tel}} = \frac{1}{2}(C_{\mathrm{tel}} - I)$$

$$S_{\mathrm{tel}} = \mathrm{c} : \varepsilon_{\mathrm{tel}}$$

which are the strain and stress tensors in the intermediate configuration, where the thermal expansion part have been removed. These quantities are independent of the magnetic field $\mathbf{H}$.

Then,

$$\varepsilon_{\mathrm{el}} = \varepsilon_{\mathrm{tel}} - C_{\mathrm{tel}}\varepsilon_{\mathrm{pzm}} + O(\|\varepsilon_{\mathrm{pzm}}\|^2)$$

and

$$J_{\mathrm{in}}W_{\mathrm{cl}} = \frac{1}{2}J_{\mathrm{th}}\varepsilon_{\mathrm{tel}} : S_{\mathrm{tel}} + J_{\mathrm{th}}W_{\mathrm{pzm}} + O(\|\varepsilon_{\mathrm{pzm}}\|^2)$$

where

$$W_{\mathrm{pzm}} = \left(\frac{1}{2}\mathrm{tr}(\varepsilon_{\mathrm{pzm}})\varepsilon_{\mathrm{tel}} - C_{\mathrm{tel}}\varepsilon_{\mathrm{pzm}}\right) : S_{\mathrm{tel}}$$

The stress is computed then as

$$S = J_{\text{th}} F_{\text{th}}^{-1}\left(S_{\text{tel}} + \frac{\partial W_{\text{pzm}}}{\partial \varepsilon_{\text{tel}}}\right) F_{\text{th}}^{-T} + S_{\text{M}} + O(\|\varepsilon_{\text{pzm}}\|^2)$$

The magnetic flux density (in the undeformed configuration) is computed as

$$\mathbf{B} = -\frac{\partial W}{\partial \mathbf{H}} = \mu_{0,\text{vac}} J C^{-1}\mathbf{H} + \mu_{0,\text{vac}}\mathbf{M} + J_{\text{th}}\frac{\partial W_{\text{pzm}}}{\partial \mathbf{H}} + O(\|\varepsilon_{\text{pzm}}\|^2)$$

where the material magnetization is given by

$$\mathbf{M} = -\frac{1}{\mu_{0,\text{vac}}}\frac{\partial W_{\text{in}}}{\partial \mathbf{H}}$$

For a magnetically linear material, it is computed as

$$\mathbf{M} = (\mu_{\text{r}} - I)\mathbf{H}$$

where μ_{r} is the relative magnetic permeability.

NONLINEAR MAGNETOSTRICTION

A commonly accepted micromagnetic description of the magnetostriction is as follows (Ref. 146):

All domains have magnetization of the same magnitude $|\mathbf{M}| = M_s$, but the magnetization can have different orientations characterized by the corresponding direction vector $\mathbf{m} = \mathbf{M}/M_s$ for each domain. The applied magnetic field changes the domain orientation.

!	In this section, the term *domain* refers to a small part of magnetic material. This is typical for micromagnetics literature, and it should not be mistaken with the concept of domain as part of the model geometry, the latter is often used in COMSOL Multiphysics documentation.

For a single crystal with cubic symmetry, the magnetostrictive strain tensor can be written as the following quadratic form using tensor notation:

$$\varepsilon_{\text{me}} = \frac{3}{2}\left[\lambda_{100}\left(\mathbf{m}\otimes\mathbf{m} - \frac{1}{3}I\right) + (\lambda_{111} - \lambda_{100})\sum_{i\neq j} m_i m_j (\mathbf{e}_i \otimes \mathbf{e}_j)\right] \qquad (3\text{-}176)$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{m}\otimes\mathbf{m})_{ij} = m_i m_j$.

Note that the magnetostrictive strain is represented by a deviatoric tensor, thus $\mathbf{tr}(\varepsilon_{me}) = 0$. This is because the deformation is related to the magnetic domain rotation, and such process should not change the material volume.

The strain in any direction given by the directional cosines β_i can be written as

$$\lambda = \frac{\Delta l}{l} = \sum_{i,j} \beta_i \beta_j \varepsilon_{ij}$$

!	The notation lambda is used for strain in this section, which is typical for micromagnetics literature. This should not be mistaken with the same notation used for stretch in other parts of the theory in COMSOL Multiphysics documentation.

Using Equation 3-176, one gets

$$\lambda = \frac{3}{2}\lambda_{100}\left(m_1^2\beta_1^2 + m_2^2\beta_2^2 + m_3^2\beta_3^2 - \frac{1}{3}\right) + 3\lambda_{111}(m_1 m_2 \beta_1\beta_2 + m_2 m_3 \beta_2\beta_3 + m_1 m_3 \beta_1\beta_3) \quad (3\text{-}177)$$

When both magnetization and measurement direction are parallel to the same crystal direction [100], one has $m_1 = \beta_1 = 1$ and all other components are zero, so that

$$\lambda = \lambda_{||} = \lambda_{100}$$

In a similar way for the [111] direction, one has for all components $m_i = \beta_i = 1/\sqrt{3}$ and $\lambda = \lambda_{111}$.

If the strain is measured in [100] direction, while all the magnetization vectors are aligned perpendicular to it, one has only the following two nonzero components: $m_2 = \beta_1 = 1$ and consequently:

$$\lambda = \lambda_{\perp} = -\frac{1}{2}\lambda_{100}$$

In many applications, such alignment of the domains is achieved by applying a compressive prestress. Thus, the maximum usable magnetostriction is achieved via a 90-degree rotation of the domains

$$\lambda_{||} - \lambda_{\perp} = \frac{3}{2}\lambda_{100}$$

For an isotropic material, $\lambda_{100} = \lambda_{111} = \lambda_s$, and Equation 3-176 becomes

$$\varepsilon_{\text{me}} = \frac{3}{2}\lambda_s\left(\mathbf{m} \otimes \mathbf{m} - \frac{1}{3}I\right)$$

For a polycrystalline material without preferred orientation, the following approximation can be used (Ref. 145):

$$\lambda_s = \frac{2}{5}\lambda_{100} + \frac{3}{5}\lambda_{111}$$

In COMSOL Multiphysics, this description of the magnetostriction is modeled using the following equation for the magnetostrictive strain:

$$\varepsilon_{\text{me}} = \frac{3}{2}\frac{1}{M_s^2}\left[\lambda_{100}\text{dev}(\mathbf{M} \otimes \mathbf{M}) + (\lambda_{111} - \lambda_{100})\sum_{i \neq j} M_i M_j (\mathbf{e}_i \otimes \mathbf{e}_j)\right] \quad (3\text{-}178)$$

The strain field is deviatoric, and Equation 3-178 exhibits the same properties as Equation 3-176 at saturation, that is, when $|\mathbf{M}| \to M_s$. Equation 3-177 is replaced by

$$\lambda = \frac{3}{2}\frac{\lambda_{100}}{M_s^2}\left[M_1^2\left(\beta_1^2 - \frac{1}{3}\right) + M_2^2\left(\beta_2^2 - \frac{1}{3}\right) + M_3^2\left(\beta_3^2 - \frac{1}{3}\right)\right] +$$
$$3\frac{\lambda_{111}}{M_s^2}(M_1 M_2 \beta_1 \beta_2 + M_2 M_3 \beta_2 \beta_3 + M_1 M_3 \beta_1 \beta_3)$$

Note that the strain vanishes when $|\mathbf{M}| \to 0$, which makes the model applicable in the whole range from full demagnetization to saturation.

For isotropic materials, the magnetostrictive strain is modeled as the following quadratic isotropic form of the magnetization field using tensor notation (Ref. 147):

$$\varepsilon_{\text{me}} = \frac{3}{2}\frac{\lambda_s}{M_s^2}\text{dev}(\mathbf{M} \otimes \mathbf{M}) \quad (3\text{-}179)$$

If the stress and strain tensors are represented by 6-component vectors using Voigt notation, the stress in the magnetostrictive material is modeled as

$$S = c_H[\varepsilon - \varepsilon_{me}(\mathbf{M})]$$

where c_H is a 6-by-6 stiffness matrix. For isotropic materials, the matrix can be represented in terms of two parameters, for example, using the Young's modulus and Poisson's ratio. Cubic materials possess only three independent components: c_{11}, c_{12}, and c_{44}.

One can derive a linear response around a given bias state characterized by a premagnetization vector $\mathbf{M}_0$. Thus,

$$\mathbf{M} = \mathbf{M}_0 + \mathbf{M}_1 = [M_{0,1}, M_{0,2}, M_{0,3}] + \mathbf{M}_1$$

$$\varepsilon_{me} = \varepsilon_{me,0}(\mathbf{M}_0) + \varepsilon_{me,1}(\mathbf{M}_0, \mathbf{M}_1)$$

where $\mathbf{M}_1$ is a perturbation.

Using Equation 3-179,one finds

$$\varepsilon_{me,1} = \frac{\lambda_s}{M_s^2}\begin{bmatrix} 2M_{0,1} & -M_{0,2} & -M_{0,3} \\ -M_{0,1} & 2M_{0,2} & -M_{0,3} \\ -M_{0,1} & -M_{0,2} & 2M_{0,3} \\ 0 & 3M_{0,3} & 3M_{0,2} \\ 3M_{0,3} & 0 & 3M_{0,1} \\ 3M_{0,2} & 3M_{0,1} & 0 \end{bmatrix}\mathbf{M}_1$$

If one assumes a unidirectional state, for example

$$\mathbf{M}_0 = [0, 0, M_0]$$

it will further simplify into

$$\varepsilon_{me,1} = \frac{\lambda_s M_0}{M_s^2}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{M}_1$$

MAGNETIZATION

The magnetization in the magnetostrictive material is found from the following nonlinear implicit relation (Ref. 148 and Ref. 149):

$$\mathbf{M} = M_s L(|\mathbf{H}_{\text{eff}}|)\frac{\mathbf{H}_{\text{eff}}}{|\mathbf{H}_{\text{eff}}|} \tag{3-180}$$

where L is the Langevin function

$$L = \coth\left(\frac{3\chi_0|\mathbf{H}_{\text{eff}}|}{M_s}\right) - \frac{M_s}{3\chi_{0m}|\mathbf{H}_{\text{eff}}|}$$

with χ_0 being the magnetic susceptibility in the initial linear region.

Other possible choices of the L function are a hyperbolic tangent, which is sometimes referred to as the Ising model

$$L = \tanh\left(\frac{\chi_0|\mathbf{H}_{\text{eff}}|}{M_s}\right)$$

and a linear function

$$L = \frac{\chi_0|\mathbf{H}_{\text{eff}}|}{M_s}$$

The latter option will make it possible to find an explicit expression for the magnetization. However, such model does not have a proper saturation behavior, and thus it should be used only in the operating range far from saturation. Both the Langevin function and hyperbolic tangent models requires the magnetization vector components to be treated as extra dependent variables.

For cubic crystals, the effective field in the material is given by

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3}{\mu_0 M_s^2}\left[\lambda_{100}\mathrm{S}_{\mathrm{d}} + (\lambda_{111} - \lambda_{100})\sum_{i \neq j}(S_{\mathrm{d}})_{ij}(\mathbf{e}_i \otimes \mathbf{e}_j)\right]\mathbf{M} \tag{3-181}$$

where $\mathbf{H}$ is the applied magnetic field, and $\otimes$ is the outer product of two vectors. The second term in Equation 3-181 represents the mechanical stress contribution to the effective magnetic field, and thus to the material magnetization, which is called the *Villari effect*. The deviatoric stress tensor is related to the strain as

$$\mathrm{S}_{\mathrm{d}} = \mathrm{dev}(\mathrm{c}_{\mathrm{H}}:(\varepsilon - \varepsilon_{\mathrm{me}}))$$

For isotropic materials, the effective magnetic field expression simplifies into

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3\lambda_s}{\mu_0 M_s^2} S_{\text{d}} \mathbf{M} \tag{3-182}$$

In addition, the magnetization and magnetic field are related to each other and to the magnetic flux density (also called the **B**-field) by

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$$

COMSOL Multiphysics solves for the magnetic vector potential **A** whose curl yields the vector **B**-field. The **H**-field is then obtained as a function of the **B**-field and magnetization

$$\mathbf{H} = \mathbf{B}/\mu_0 - \mathbf{M}$$

The effective tangent piezomagnetic coupling coefficients can be computed as

$$\text{d} = \frac{\partial \varepsilon_{\text{me}}}{\partial \mathbf{H}}$$

For an isotropic material, the derivative can be evaluated to give

$$\text{d} = 3\frac{\lambda_s}{M_s^2}\left(\mathbf{M} \otimes \chi_{\text{m}} - \frac{1}{3}(I \otimes I) : (\mathbf{M} \otimes \chi_{\text{m}})\right)$$

where

$$\chi_{\text{m}} = \frac{\partial \mathbf{M}}{\partial \mathbf{H}}$$

is the tangent magnetic susceptibility, and $\otimes$ is the outer product of two tensors, $(\mathbf{A} \otimes \mathbf{B})_{ijkl} = A_{ij} B_{kl}$. The corresponding expression in case of cubic material is more complicated, but it has a similar structure involving products of the magnetization and tangent susceptibility components. An important observation from the above formulas is that the piezomagnetic coefficients should reach their maximum (or minimum) at certain strength of the applied bias field if the saturation effect is taken into account. This is because **M** is zero at zero applied field, while χ_{m} tends to zero at large applied field magnitudes because of saturation.

The piezomagnetic coupling tensor $\mathbf{d}$ is a third-order tensor. Due to the symmetry, it can be conventionally represented by a 3-by-6 matrix d_{HT} with only few nonzero components.

HYSTERESIS MODELING

The Jiles–Atherton hysteresis model for magnetostrictive materials is available COMSOL Multiphysics. The model assumes that the total magnetization can be represented as a sum of hysteretic and anhysteretic parts, the latter one is given by Equation 3-180, thus

$$\mathbf{M}_{an} = M_s L(|\mathbf{H}_{eff}|)\frac{\mathbf{H}_{eff}}{|\mathbf{H}_{eff}|}$$

Compared to Equation 3-181 and Equation 3-182, the effective magnetic field $\mathbf{H}_{eff}$ gets one more term $\alpha\mathbf{M}$, where α is the interdomain coupling parameter.

The change in the total magnetization caused by the change on the effective magnetic field is represented as

$$\begin{aligned} d\mathbf{M} &= c_r d\mathbf{M}_{an} + \max(\xi \cdot d\mathbf{H}_{eff}, 0)\xi/|\xi| \\ \xi &= k_p^{-1}(\mathbf{M}_{an} - \mathbf{M}) \end{aligned} \tag{3-183}$$

where c_r is the reversibility parameter, and k_p is the pining loss parameter.

Equation 3-183 can be solved using either a time-dependent analysis or a stationary parametric sweep.

For more details, see The Jiles–Atherton Hysteresis Model in the *AC/DC Module User's Guide*.

Electrostriction

Electrostriction is an interaction in solid dielectric materials, where an electric field applied on the material generates the deformation of the material (direct effect), and a mechanics stress applied on it changes the material polarization (inverse effect).

The direct electrostrictive effect for a material of arbitrary symmetry can be represented as the following additive contribution to the strain (Ref. 153):

$$\varepsilon_{em} = \mathbf{Q} : (\mathbf{P} \otimes \mathbf{P})$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{P}\otimes\mathbf{P})_{ij} = P_i P_j$. This expression is quadratic in polarization $\mathbf{P}$ in contrast to linear piezoelectricity.

The fourth order tensor $\mathbf{Q}$ can be effectively represented by a 6-by-6 coupling matrix. Further simplification due to material symmetry is possible in most cases. For example, for piezoelectric ceramics, such coupling matrix can be characterized by three independent components: Q_{11}, Q_{12}, and Q_{44}.

If the polarization in the material is linear with the applied electric field, the electrostrictive strain can be written equivalently in terms of the electric field:

$$\varepsilon_{\mathrm{em}} = \mathrm{M} : (\mathbf{E} \otimes \mathbf{E})$$

where

$$\mathrm{M} = \varepsilon_{0,\,\mathrm{vac}}^{2}\mathbf{Q} : (\chi \otimes \chi)$$

where $\varepsilon_{0,\mathrm{vac}}$ is the electric permittivity of free space, and χ is the linear electric susceptibility tensor (measured at zero mechanical deformation).

The total polarization in the material can then be written as

$$\mathbf{P} = \varepsilon_{0,\,\mathrm{vac}}\chi\mathbf{E} + 2(\mathrm{M} : \mathrm{S})\mathbf{E} \tag{3-184}$$

where the mechanical stress in the material is computed assuming a mechanically linear material as

$$\mathrm{S} = \mathsf{C} : (\varepsilon - \varepsilon_{\mathrm{em}})$$

where C is the fourth order elasticity tensor, and the strain tensor is given by

$$\varepsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{T})$$

The last term in Equation 3-184 represents the inverse electrostrictive effect.

The fourth order tensor $\mathbf{Q}$ can be effectively represented by a 6-by-6 coupling matrix. Further simplification due to material symmetry is possible in most cases. Thus, for cubic crystal material, the coupling matrix can be characterized by three independent components: Q_{11}, Q_{12}, and Q_{44}.

For an isotropic material, only two independent coefficients Q_{11} and Q_{12} are needed since $Q_{44} = 0{,}5(Q_{11} - Q_{12})$. Two corresponding M-constants can be computed as

$$M_{ij} = (\varepsilon_{0,\,\mathrm{vac}}\chi_0)^2 Q_{ij}$$

where χ_0 denotes the only independent component of the electric susceptibility tensor.

For isotropic materials, the fourth order elasticity tensor C has only two independent components. The most common choice to represent those are by specifying the Young's modulus E_{YM} and Poisson's ratio ν.

The following alternative electrostrictive parameters are defined in Ref. 151 for a linear isotropic material:

$$a_1 = 2(\varepsilon_{0,\,\mathrm{vac}}\chi)^2 E_{\mathrm{YM}} \frac{(1-\nu)Q_{11} + 2\nu Q_{12}}{(1+\nu)(1-2\nu)}$$

$$a_2 = 2(\varepsilon_{0,\,\mathrm{vac}}\chi)^2 E_{\mathrm{YM}} \frac{\nu Q_{11} + Q_{12}}{(1+\nu)(1-2\nu)}$$

One more commonly used alternative definition is that introduced in Ref. 152. Using a_1' and a_2' for the constants used in Ref. 152, one has $a_1' = a_1 - a_2$ and $a_2' = a_2$.

The SI units for different electrostrictive parameters are summarized in Table 3-7.

TABLE 3-7: UNITS FOR ELECTROSTRICTIVE CONSTANTS.

MATERIAL PARAMETER	UNIT (SI)
Q	m^2/V^2
M	m^4/C^2
a	F/m

Ferroelectroelasticity

The ferroelectroelasticity and ferroelectricity phenomena are related to phase transitions in materials. In its ferroelectric phase, the material exhibits spontaneous polarization, so that it is constituted of domains with nonzero polarization even at zero applied field. Electrostriction in ferroelectroelastic materials can be related to the domain rotation. Thus, the applied electric field can both rearrange the domains resulting into the net polarization and rotate the domains mechanically. Thus, the material extends in the direction of the electric field and contracts in the direction perpendicular to the field. The domain rotation can be affected by an applied mechanical stress, which also results into the effective polarization. At very large electric fields, the electrostrictive effect saturates, as all ferroelectric domains in the material are aligned along the direction of the applied field. Domain wall interactions can also lead to a significant hysteresis in the polarization and strain.

The direct electrostrictive effect for a material of arbitrary symmetry can be represented as the following additive contribution to the strain (Ref. 154):

$$\boldsymbol{\varepsilon}_{\text{em}} = \mathbf{Q} : (\mathbf{P} \otimes \mathbf{P})$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{P}\otimes\mathbf{P})_{ij} = P_i P_j$, so $\boldsymbol{\varepsilon}_{\text{em}}$ is quadratic with respect to the polarization $\mathbf{P}$. Due to the symmetry, the fourth order tensor $\mathbf{Q}$ can be effectively represented by a 6-by-6 coupling matrix. The number of independent components in the matrix depends on the material symmetry. For example, for piezoelectric ceramics, the matrix can be characterized by three independent components: Q_{11}, Q_{12}, and Q_{44}.

For ferroelectroelastic materials, the polarization vector is nonlinear function of the electric field and possible mechanical stress in the material.

If hysteresis in the material can be neglected, the polarization is computed from the implicit relation

$$\mathbf{P} = \mathbf{P}_{\text{an}}(\mathbf{E}, \mathbf{P})$$

where the anhysteretic polarization function is assumed to have the following special form:

$$\mathbf{P}_{\text{an}} = P_{\text{s}} L(|\mathbf{E}_{\text{eff}}|) \frac{\mathbf{E}_{\text{eff}}}{|\mathbf{E}_{\text{eff}}|} \tag{3-185}$$

where P_{s} is the saturation polarization.

The polarization shape is characterized by the function L with the following properties:

For weak effective fields, the polarization is nearly linear

$$\mathbf{P} \approx \chi_0 \mathbf{E}_{\text{eff}}$$

and can be characterized by the initial electric susceptibility matrix χ_0.

For strong fields, the polarization magnitude approaches the saturation value

$$\mathbf{P} \approx P_{\text{s}} \frac{\mathbf{E}_{\text{eff}}}{|\mathbf{E}_{\text{eff}}|}$$

Two possible choices are the Langevin function

$$L = \coth\left(\frac{3\chi_0|\mathbf{E}_{\text{eff}}|}{P_{\text{s}}}\right) - \frac{P_{\text{s}}}{3\chi_0|\mathbf{E}_{\text{eff}}|}$$

and a hyperbolic tangent (Ising spin model):

$$L = \tanh\left(\frac{\chi_0|\mathbf{E}_{\text{eff}}|}{P_s}\right)$$

The effective electric field is given by

$$\mathbf{E}_{\text{eff}} = \mathbf{E} + \alpha\mathbf{P} + 2(\mathrm{S} : \mathrm{Q})\mathbf{P} \tag{3-186}$$

where $\mathbf{E}$ is the applied electric field, α is a material parameter called the inter-domain coupling, and the mechanics stress is computed assuming mechanically linear material as

$$\mathrm{S} = \mathsf{C} : (\varepsilon - \varepsilon_{\text{em}})$$

where C is the fourth order elasticity tensor. The last term in Equation 3-186 represents the inverse electrostrictive effect.

The effective tangent piezoelectric coupling tensor can be computed as

$$\mathrm{d} = \frac{\partial \varepsilon_{\text{em}}}{\partial \mathbf{E}} = 2\varepsilon_{0,\,\text{vac}}\mathrm{Q} : (\mathbf{P} \otimes \chi_{\text{e}})$$

where $\varepsilon_{0,\text{vac}}$ is the electric permittivity of free space, $\otimes$ is the outer product of a vector and a matrix, and

$$\chi_{\text{e}} = \frac{1}{\varepsilon_{0,\,\text{vac}}}\frac{\partial \mathbf{P}}{\partial \mathbf{E}}$$

is the tangent electric susceptibility matrix. The piezoelectric coupling tensor d is a third-order tensor. Due to the symmetry, it can be conventionally represented by a 3-by-6 matrix with only few nonzero components. An important observation from the above formula is that the piezoelectric coefficients should reach their maximum (or minimum) at certain strength of the applied bias field. This is because $\mathbf{P}$ is zero at zero applied field, while χ_{e} tends to zero at large applied field magnitudes because of saturation.

If the mechanical deformation due to the electrostriction is assumed to be volume preserving, the following form can be used for the electrostrictive strain for isotropic materials:

$$\varepsilon_{em} = \frac{3}{2}\frac{\lambda_s}{P_s^2}\mathrm{dev}(\mathbf{P}\otimes\mathbf{P})$$

where λ_s is the saturation electrostriction, and the deviatoric part can be computed for any matrix A as

$$\mathrm{dev}(A) = A - \frac{1}{3}(I\otimes I):\mathrm{A}$$

The stress in Equation 3-186 is replaced then by

$$S_d = \mathrm{dev}(\mathrm{C} : (\varepsilon - \varepsilon_{em}))$$

and tangent piezoelectric coupling tensor becomes

$$\mathrm{d} = 3\frac{\varepsilon_{0,\,vac}\lambda_s}{M_s^2}\left(\mathbf{P}\otimes\chi_e - \frac{1}{3}(I\otimes I):(\mathbf{P}\otimes\chi_e)\right)$$

HYSTERESIS MODELING

The Jiles–Atherton model for ferroelectric hysteresis assumes that the total polarization can be represented as a sum of reversible and irreversible parts. The polarization change is computed from the following incremental equation:

$$d\mathbf{P} = \mathrm{c_r}d\mathbf{P}_{an} + (\mathrm{I} - \mathrm{c_r})d\mathbf{P}_{irr} \tag{3-187}$$

where the anhysteretic polarization is found using Equation 3-185, and the irreversible polarization change is computed as

$$d\mathbf{P}_{irr} = \max(\zeta\cdot d\mathbf{E}_{eff}, 0)\frac{\zeta}{|\zeta|}$$

$$\zeta = \mathrm{k_p^{-1}}(\mathbf{P}_{an} - \mathbf{P}_{irr})$$

where the pinning loss is characterized by the parameter $\mathrm{k_p}$.

Equation 3-187 can be solved using either a time-dependent analysis or a stationary parametric sweep.

Rigid Material

A rigid domain, or a rigid body, is an idealization of a body in which the deformation is neglected. In other words, the distance between any two given points of a rigid body remains constant in time regardless of any external forces acting on it. An object can be assumed to be perfectly rigid if its flexibility can be neglected in comparison with other flexibilities in the system, and when there is no need to compute the stress in the object. To model a rigid domain, you use the **Rigid Material** material model.

Rigid Material is available as a material model in the Solid Mechanics, Shell, Beam, and Multibody Dynamics interfaces. This theory section applies to all interfaces. There are some minor differences between the versions of the **Rigid Material** which will be described as they appear.

In the following, when talking about rigid domains, the geometrical object can be a domain, a boundary, or an edge, depending on physics interface and spatial dimension.

A rigid domain can consist of a selection of several geometrical domains. These domains will act as a single rigid object, irrespective of whether they are geometrically connected or not.

Adjacent geometrical objects selected in different **Rigid Material** nodes are independent, and can even penetrate each other.

The **Rigid Material** is a material model, which is mutually exclusive to all other material models. The only material property needed is the mass density.

RIGID DOMAIN KINEMATICS

When a body is rigid, it is sufficient to describe the motion of at least three not collinear particles. It is then possible to reconstruct the motion of all other particles in the body. Usually, a mathematically more convenient, but equivalent, approach is used. The motion of the whole body is represented by

- The linear motion of the body. The motion of one of the particles of the body, chosen as a reference point (often coinciding with the center of mass).
- The angular motion (also known as orientation or attitude) of the body.

The degrees of freedom needed to represent the linear and angular motion are known as rigid body translation and rigid body rotation degrees of freedom.

In 2D axisymmetric interfaces, a rigid domain has only a single degree of freedom: translation along the Z axis. The following theory sections only describe the more general cases. The axisymmetric formulation is a trivial specialization where all rotational parts of expressions are dropped.

In 2D, this is represented by two in-plane translations and the scalar rotation around the z-axis.

In 3D the situation is more complex. Six degrees of freedom are necessary. They are usually selected as three translations and three parameters for the rotation. For finite rotations, any choice of three rotation parameters is will give a singularity at some specific set of angles. For this reason, a four-parameter *quaternion* representation is used for the rotations in COMSOL Multiphysics. Thus, each rigid domain in 3D actually has seven degrees of freedom: three for the translation, and four for the rotation. The quaternion parameters are called a, b, c, and d. These four parameters are not independent, so an extra equation stating that

$$a^2 + b^2 + c^2 + d^2 = 1$$

is added.

The connection between the quaternion parameters and the rotation matrix R is

$$R = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 \end{bmatrix}$$

For the geometrically linear case, the quaternion constraint and the rotation matrix definition are reduced to:

$$R = \begin{bmatrix} 1 & -2d & 2c \\ 2d & 1 & -2b \\ -2c & 2b & 1 \end{bmatrix}$$

$$a = 1$$

In 2D, the rotational degree of freedom is the angle of rotation about the z-axis ϕ, and its relation with the rotation matrix R is

$$R = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For the geometrically linear case, the 2D rotation matrix is reduced to:

$$R = \begin{bmatrix} 1 & -\phi & 0 \\ \phi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Under translation and rotation of a rigid domain, the complete expression for the displacement of any point on the rigid body is given by

$$\mathbf{u}_{\mathrm{d}} = \mathbf{u} + (R - I) \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}})$$

where $\mathbf{X}$ are the material coordinates of any point in the rigid domain, $\mathbf{X}_{\mathrm{M}}$ is the center of mass of the rigid domain, $\mathbf{u}$ is the translation vector at the center of mass, and I is the identity matrix.

The rigid body displacement at the center of mass ($\mathbf{u}$) are degrees of freedom. Thus, the rigid body translational velocity and acceleration can be evaluated by directly taking the time derivatives of $\mathbf{u}$. In the time domain it can be expressed as

$$\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t} \qquad \ddot{\mathbf{u}} = \frac{\partial}{\partial t}\dot{\mathbf{u}}$$

In the frequency domain, they can be expressed in terms of frequency (ω):

$$\dot{\mathbf{u}} = (i\omega)\mathbf{u} \qquad \ddot{\mathbf{u}} = (i\omega)\dot{\mathbf{u}} = -\omega^2\mathbf{u}$$

The same is true for the rotation in 2D since the rigid body rotation ϕ is the degree of freedom. The rigid body angular velocity and acceleration can be evaluated by directly taking the time derivatives of ϕ.

In 3D, the situation is different and the total rotation of the rigid domain can be presented as a function of quaternion:

$$\Theta = \frac{2\,\mathrm{atan}\left(\frac{\sqrt{b^2+c^2+d^2}}{a}\right)}{\sqrt{b^2+c^2+d^2}}\begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

The parameter a can be considered as measuring the rotation, while b, c, and d can be interpreted as the orientation of the rotation vector. For small rotations, this relation simplifies to:

$$\Theta = 2\begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

The angular velocity of the rigid domain is computed as

$$\dot{\Theta} = \begin{bmatrix} Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \qquad Q = 2(\bar{q} \otimes \dot{q}) \qquad q = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Here $\bar{q}$ is the conjugate of q, and $\otimes$ denotes quaternion multiplication.

The angular acceleration of the rigid domain can be evaluated by taking the time derivative of the angular velocity.

RIGID DOMAIN DYNAMICS

The governing equation for a rigid domain can be written as a balance between the inertial (internal) forces and applied external forces. A rigid domain has only one internal force, the inertial force. This means that only the mass density of a domain is required to define the rigid domain material model.

The inertial forces and inertial moments about the center of mass are:

$$\mathbf{F} = m\ddot{\mathbf{u}} \qquad \mathbf{M} = I\ddot{\Theta}$$

where $\ddot{\mathbf{u}}$ and $\ddot{\Theta}$ are the linear and angular accelerations of a rigid domain.

The inertial properties mass (m) and moment of inertia tensor (I) of a rigid domain are computed as

$$m = \int \rho dV$$

$$\mathbf{X}_{\mathrm{M}} = \frac{\int \rho \mathbf{X} dV}{m}$$

$$I = \int (((\mathbf{X} - \mathbf{X}_{\mathrm{M}})^{T} \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}}))E_3 - (\mathbf{X} - \mathbf{X}_{\mathrm{M}}) \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}})^{T})\rho dV$$

where E_3 and $\mathbf{X}_{\mathrm{M}}$ are the identity matrix and the center of mass of a rigid domain, respectively. The special case for the Shell interface is described in Rigid Domain for Shells.

In 2D, the expressions for inertial forces, inertial moments, and moment of inertia reduce to:

$$\mathbf{F} = m\ddot{\mathbf{u}} \qquad M = I_z \ddot{\phi}$$

$$I_z = \int ((\mathbf{X} - \mathbf{X}_{\mathrm{M}}) \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}}))\rho h \, dA$$

where the volume integration has been replaced by an area integration multiplied by the out-of-plane thickness h.

The equations of motion for the rigid domain are

$$m\ddot{\mathbf{u}} + \sum \mathbf{F}_{\mathrm{I}} = \sum \mathbf{F}_{\mathrm{ext}}$$

and

$$RIR^{T}\ddot{\Theta} + \dot{\Theta} \times (RIR^{T}\dot{\Theta}) + \sum \mathbf{M}_{\mathrm{I}} = \sum \mathbf{M}_{\mathrm{ext}}$$

Here, the subscripts I and ext denote inertial and external forces, respectively, and R is the current rotation matrix. The inertial forces are contributions from Mass and Moment of Inertia nodes.

In 2D, the moment equations are simplified to the scalar equation

$$I_z \ddot{\varphi} + \sum M_{\mathrm{I}} = \sum M_{\mathrm{ext}}$$

INITIAL VALUE

The **Rigid Material** is a separate material model, and it overrides the default **Linear Elastic Material** node and the default **Initial Values** node in the physics interface. The initial values are given in a separate **Initial Values** subnode for each **Rigid Material**.

In the Multibody Dynamics interface version of **Rigid Material**, it is also possible to get initial values for all domains from the interface level **Initial Values** section. This is the default option. The **Initial Values** subnode is only present under **Rigid Material** if **Locally defined** has been selected.

If many rigid domains are present in a system and they have the same initial values, then it is often better to define initial values at the interface level once and to reference these in all features.

The initial values for the rigid body translation, rigid body rotation, and the first time derivatives can be prescribed about any point — a center of rotation — in a selected coordinate system. The center of rotation can be defined using

- The center of mass of the rigid domain
- Global coordinates of the center of rotation
- The centroid of a set of selected entities (boundaries, edges, or points)

Given the initial values of translation ($\mathbf{u}$), rotation (ϕ), translational velocity ($\frac{\partial \mathbf{u}}{\partial t}$) and angular velocity ($\omega$) about a center of rotation ($\mathbf{X}_c$), the rigid body displacement and quaternion degrees of freedom are initialized as

$$\mathbf{u}_0 = \mathbf{u} + \mathbf{u}_\mathrm{r}$$
$$\mathbf{u}_\mathrm{r} = (-\mathbf{r} + \hat{\Omega}(\hat{\Omega} \cdot \mathbf{r}))(1 - \cos\phi) + (\hat{\Omega} \times \mathbf{r})\sin\phi$$
$$\mathbf{r} = \mathbf{X}_\mathrm{M} - \mathbf{X}_\mathrm{c}$$
$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)_0 = (\omega \times (\mathbf{r} + \mathbf{u}_\mathrm{r})) + \frac{\partial \mathbf{u}}{\partial t}$$
$$\begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0 = \begin{bmatrix} \cos\frac{\phi}{2} \\ \sin\frac{\phi}{2}\hat{\Omega} \end{bmatrix}$$
$$\begin{bmatrix} \frac{\partial a}{\partial t} \\ \frac{\partial \mathbf{b}}{\partial t} \end{bmatrix}_0 = \frac{1}{2}\left(\begin{bmatrix} 0 \\ \omega \end{bmatrix} \otimes \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0\right)$$

In 2D, the expressions for the initial values reduce to:

$$\mathbf{u}_0 = \mathbf{u} + \mathbf{u}_\mathrm{r}$$
$$\mathbf{u}_\mathrm{r} = -\mathbf{r}(1 - \cos\phi) + (\mathbf{e}_z \times \mathbf{r})\sin\phi$$
$$\mathbf{r} = \mathbf{X}_\mathrm{M} - \mathbf{X}_\mathrm{c}$$
$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)_0 = (\mathbf{e}_z \times (\mathbf{r} + \mathbf{u}_\mathrm{r}))\frac{\partial \phi}{\partial t} + \frac{\partial \mathbf{u}}{\partial t}$$
$$\phi_0 = \phi$$
$$\left(\frac{\partial \phi}{\partial t}\right)_0 = \frac{\partial \phi}{\partial t}$$

The variable $\mathbf{u}_\mathrm{r}$ is the translation at the center of mass due to a rotation around the center of rotation, and is thus zero when the two points coincide. In the case that you are entering the data using a separate center of rotation, you must pay special attention to how the initial displacement and velocity are composed if initial rotations and rotational velocities are present.

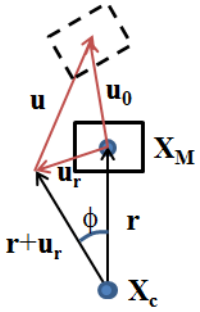


Figure 3-35: Initial displacement of a rigid body.

MASS AND MOMENT OF INERTIA

Sometimes a rigid domain needs the added effect of an associated abstract rigid object, which is physically not modeled and where the inertial properties are known. You can model this using **Mass and Moment of Inertia**, where the inertial properties of this abstract domain (center of mass, mass, and moment of inertia tensor) can be directly entered.

The formulation for an abstract rigid object is similar to the physical rigid domain with these exceptions:

- The inertial properties of the abstract rigid object are input by the user instead of being computed from a physical domain.
- No extra degrees of freedom are created. The inertial forces generated by this feature will be computed based on the distance from the center of gravity of the rigid domain to which it belongs, and the values of the degrees of freedom there.

The inertial force contributions are

$$\mathbf{F}_{\mathrm{I}} = m\frac{d^2}{dt^2}(\mathbf{u} + (R - E_3)\mathbf{X}_{\mathrm{mc}})$$

where $\mathbf{X}_{\mathrm{mc}}$ is the vector from the center of mass of the rigid domain ($\mathbf{X}_{\mathrm{M}}$) to the center of mass of this contribution ($\mathbf{X}_{\mathrm{m}}$),

$$\mathbf{X}_{\mathrm{mc}} = \mathbf{X}_{\mathrm{m}} - \mathbf{X}_{\mathrm{M}}$$

The inertial moment contributions are

$$\mathbf{M}_{\mathrm{I}} = RIR^{T}\ddot{\Theta} + \dot{\Theta} \times (RIR^{T}\dot{\Theta}) + \mathbf{X}_{\mathrm{mc}} \times \mathbf{F}_{\mathrm{I}}$$

In 2D, there is only a scalar moment contribution:

$$M_{\mathrm{I}} = I_z\ddot{\varphi} + (\mathbf{X}_{\mathrm{mc}} \times \mathbf{F}_{\mathrm{I}}) \cdot \mathbf{e}_z$$

CONSTRAINTS

The constraints for a rigid domain are different in nature than those applied to flexible domains. In a flexible domain, a constraint can be applied at various entity levels: domains, boundaries, edges, or points. Since the degrees of freedom of the rigid domain are global and present only at the center of mass, boundary conditions are used to constrain these global degrees of freedom, which is why a global selection is needed.

The constraints used for a flexible domain, for example **Fixed Constraint**, **Prescribed Displacement**, **Rigid Connector**, or **Attachment**, are not applicable to a rigid domain.

In a rigid domain the **Prescribed Displacement/Rotation** or **Fixed Constraint** subnode is used instead to constrain its degrees of freedom.

The **Prescribed Displacement/Rotation** node can be used to:

- Prescribe the displacement components in arbitrary directions at a given point.
- Constrain rotations in arbitrary directions.
- Prescribe a nonzero rotation around an arbitrary axis.

The displacement and rotation can be prescribed in a selected coordinate system about an arbitrary center of rotation. The center of rotation can be defined using

- The center of mass of the rigid domain.
- Global coordinates of the center of rotation.
- The centroid of a set of selected entities (boundaries, edges, or points).

The displacement at the center of rotation is computed as

$$\mathbf{u}_{\mathrm{c}} = \mathbf{u} + (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X}_{\mathrm{c}} - \mathbf{X}_{\mathrm{M}}) \tag{3-188}$$

The components of this displacement vector are prescribed individually in the selected coordinate system. Through Equation 3-188, a constraint on a translation will impose

a relation between translational and rotational degrees of freedom if the center of rotation differs from the center of mass.

To prescribe the rotation in 3D, the imaginary part of the quaternion is prescribed as

$$\mathbf{b} = \hat{\Omega}\sin\frac{\phi_0}{2}$$

where $\hat{\Omega}$ and ϕ_0 are the axis of rotation and angle of rotation, respectively.

In 2D, the out-of-plane rotation angle is directly constrained to the prescribed value of the rotation.

LOADS

The loads available for a flexible domain can also be used for a rigid domain. In addition to these boundary conditions, a rigid domain also has global subnodes for applying forces and moments. If you use **Applied Force**, a force and its location can be prescribed in a selected coordinate system. A force implicitly also contributes to the moment unless it is applied at the center of mass of a rigid domain. If an **Applied Moment** node is used, a moment can be prescribed in a selected coordinate system.

CONNECTING TO OTHER BODIES

When a rigid domain and a flexible domain share a boundary (Shell: edge, Beam: point), the connection is automatic. All displacements on the flexible domain are controlled by the degrees of freedom of the rigid domain, so that

$$\mathbf{u}_{\text{flex}} = (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X} - \mathbf{X}_{\text{M}}) + \mathbf{u}$$

where $\mathbf{X}$ is a coordinate on the boundary. If rotational degrees of freedom are present, which is the case in the Shell and Beam interfaces, the rotations are set equal to those of the rigid domain.

In the Multibody Dynamics interface, a rigid domain can be also be connected to another rigid or flexible domain using joints. A rigid component can be directly selected in the joints to establish a connection. This stands in contrast to a flexible component, which needs an attachment.

Adjacent domains selected in different **Rigid Material** nodes are independent objects.

Safety Factor Evaluation

There are many theories for predicting material failure. These can predict, for instance whether a ductile material will yield or not, or if a brittle material will crack under a given set of loads.

Tsai and Wu (Ref. 97, Ref. 110) proposed a stress-dependent criterion intended at modeling failure in composites. Under the Tsai–Wu criterion, failure occurs when a given quadratic function of stress is greater than zero. The failure criterion is given by

$$g(\sigma) = \sigma : (\boldsymbol{F}\sigma) + \boldsymbol{f} : \sigma - 1$$

where, σ is the stress tensor, $\boldsymbol{F}$ a fourth rank tensor (SI unit: $1/\mathrm{Pa}^2$) and $\boldsymbol{f}$ is a second rank tensor (SI unit: $1/\mathrm{Pa}$). For the Tsai–Wu criterion, failure occurs when $g(\sigma) \geq 0$.

Due to the symmetry of these tensors, the fourth rank tensor can be represented by a symmetric 6-by-6 matrix, and the second rank tensor by a 6-by-1 vector (see Voigt order in the section Tensor vs. Matrix Formulations).

Certain constraints ensure that the failure surface $g(\sigma) = 0$ forms a closed ellipsoid in the stress space. Also, thermodynamic considerations restrict the value of some components of the fourth rank tensor. These restrictions are summarized as (no index summation)

$$\boldsymbol{F}_{ii} > 0 \quad \text{and} \quad \boldsymbol{F}_{ii}\boldsymbol{F}_{jj} \geq \boldsymbol{F}_{ij}^2$$

The *failure index* is computed from the failure criterion as

$$fi = g(\sigma) + 1$$

so failure is predicted for a failure index greater than one, $fi \geq 1$.

The *damage index* is given by a Boolean expression based on the failure criterion

$$di = \begin{cases} 1 & g(\sigma) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

here $di = 1$ means damage, and $di = 0$ represents a healthy material.

The *safety factor*, s_f, also called *reserve factor* or *strength ratio*, is computed by scaling the stress tensor such as the failure criterion is equal to zero

$$g(s_\mathrm{f}\sigma) = 0$$

For a quadratic failure criterion, as the Tsai–Wu criterion, this means solving a quadratic equation for the safety factor variable s_f

$$s_f^2(\sigma:(F\sigma)) + s_f(f:\sigma) - 1 = 0$$

the safety factor is then obtained from the smallest positive root.

For an isotropic criterion, such as the von Mises criterion, $g(\sigma) = \sigma_{mises}/\sigma_{ts} - 1$, the safety factor is given by $s_f = \sigma_{ts}/\sigma_{mises}$.

The *margin of safety* (Ref. 110) is then computed from the safety factor s_f as

$$ms = s_f - 1$$

Following the Tsai–Wu formalism, different orthotropic criteria can be defined by setting appropriate values for the coefficients in the tensors F and f.

Use the **Safety** subnode to set up variables which can be used to check the risk of failure according to various criteria. It can be used in combination with Linear Elastic Material, Linear Elastic Material, Layered, or Nonlinear Elastic Materials.

Add any number of **Safety** subnodes to a single material model; the contents of these features will not affect the analysis results because they do not account for postfailure analysis.

Add **Safety** subnodes after having performed an analysis and just do an **Update Solution** in order to access to the new variables for results evaluation.

Add a **Safety** subnode to calculate the safety factor for a specified criterion. When multiple **Safety** subnodes are added, additional built-in variables become available. These additional variables—*combined safety factor*, *combined failure index*, *combined damage index*, and *combined margin of safety*—offer a conservative approach to safety assessment.

ANISOTROPIC TSAI–WU CRITERION

For this anisotropic criterion, enter 21 coefficients to define the 6-by-6 matrix F, and six coefficients to define the vector f. The failure criterion is evaluated from the expression

$$g(\sigma) = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^T \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\ F_{12} & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\ F_{13} & F_{23} & F_{33} & F_{34} & F_{35} & F_{36} \\ F_{14} & F_{24} & F_{34} & F_{44} & F_{45} & F_{46} \\ F_{15} & F_{25} & F_{35} & F_{45} & F_{55} & F_{56} \\ F_{16} & F_{26} & F_{36} & F_{46} & F_{56} & F_{66} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}^T \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} - 1$$

here, σ_{ij} are the stress tensor components given in the local coordinate system of the parent node.

ORTHOTROPIC TSAI–WU CRITERION

For this orthotropic criterion, enter nine coefficients corresponding to the tensile strengths σ_{tsi}, compressive strengths σ_{csi}, and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The Tsai–Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{cs1}\sigma_{ts1}}, F_{22} = \frac{1}{\sigma_{cs2}\sigma_{ts2}}, F_{33} = \frac{1}{\sigma_{cs3}\sigma_{ts3}},$$

$$F_{44} = \frac{1}{\sigma_{ss23}^2}, F_{55} = \frac{1}{\sigma_{ss13}^2}, F_{66} = \frac{1}{\sigma_{ss12}^2},$$

$$F_{12} = -\frac{1}{2}\sqrt{F_{11}F_{22}}, F_{13} = -\frac{1}{2}\sqrt{F_{11}F_{33}}, F_{23} = -\frac{1}{2}\sqrt{F_{22}F_{33}}$$

$$f_1 = \frac{1}{\sigma_{ts1}} - \frac{1}{\sigma_{cs1}}, f_2 = \frac{1}{\sigma_{ts2}} - \frac{1}{\sigma_{cs2}}, f_3 = \frac{1}{\sigma_{ts3}} - \frac{1}{\sigma_{cs3}}$$

all the other coefficients in F and f are set to zero.

In the plane stress version of the Tsai–Wu orthotropic criterion, the only nonzero coefficients are

$$F_{11} = \frac{1}{\sigma_{cs1}\sigma_{ts1}}, F_{22} = \frac{1}{\sigma_{cs2}\sigma_{ts2}}, F_{66} = \frac{1}{\sigma_{ss12}^2}, F_{12} = -\frac{1}{2}\sqrt{F_{11}F_{22}}$$

$$f_1 = \frac{1}{\sigma_{ts1}} - \frac{1}{\sigma_{cs1}}, f_2 = \frac{1}{\sigma_{ts2}} - \frac{1}{\sigma_{cs2}}$$

TSAI–HILL CRITERION

For this orthotropic criterion, enter six coefficients corresponding to the tensile strengths σ_{tsi} and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The equivalent coefficients for the Anisotropic Tsai–Wu Criterion are then computed from

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0 \text{, or } F_{11} = \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts2}^2} \quad \text{for } \sigma_{22} > 0 \text{, or } F_{22} = \frac{1}{\sigma_{cs2}^2} \quad \text{for } \sigma_{22} < 0$$

$$F_{33} = \frac{1}{\sigma_{ts3}^2} \quad \text{for } \sigma_{33} > 0 \text{, or } F_{33} = \frac{1}{\sigma_{cs3}^2} \quad \text{for } \sigma_{33} < 0$$

$$F_{44} = \frac{1}{\sigma_{ss23}^2}, \; F_{55} = \frac{1}{\sigma_{ss13}^2}, \; F_{66} = \frac{1}{\sigma_{ss12}^2},$$

$$F_{12} = -\frac{1}{2}(F_{11} + F_{22} - F_{33}), \; F_{13} = -\frac{1}{2}(F_{11} + F_{33} - F_{22}),$$

$$F_{23} = -\frac{1}{2}(F_{22} + F_{33} - F_{11})$$

all the other coefficients in F and f tensors are set to zero. See also Hill Criterion.

In the plane stress version of the Tsai–Hill criterion, some terms become zero

$$F_{33} = F_{13} = F_{23} = F_{55} = F_{44} = 0$$

and the F_{12} term is modified as follows

$$F_{12} = \begin{cases} -\dfrac{1}{2\sigma_{ts1}^2} & \text{for } \operatorname{sign}(\sigma_{11}) = \operatorname{sign}(\sigma_{22}) \\ -\dfrac{1}{2\sigma_{tc1}^2} & \text{for } \operatorname{sign}(\sigma_{11}) \neq \operatorname{sign}(\sigma_{22}) \end{cases}$$

HOFFMAN CRITERION

For this orthotropic criterion, enter nine coefficients corresponding to the tensile strengths σ_{tsi}, compressive strengths σ_{csi}, and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The equivalent coefficients for the Anisotropic Tsai–Wu Criterion are then computed from

$$F_{11} = \frac{1}{\sigma_{cs1}\sigma_{ts1}}, F_{22} = \frac{1}{\sigma_{cs2}\sigma_{ts2}}, F_{33} = \frac{1}{\sigma_{cs3}\sigma_{ts3}},$$

$$F_{44} = \frac{1}{\sigma_{ss23}^2}, F_{55} = \frac{1}{\sigma_{ss13}^2}, F_{66} = \frac{1}{\sigma_{ss12}^2},$$

$$F_{12} = -\frac{1}{2}(F_{11} + F_{22} - F_{33}), F_{13} = -\frac{1}{2}(F_{11} + F_{33} - F_{22}),$$

$$F_{23} = -\frac{1}{2}(F_{22} + F_{33} - F_{11})$$

$$f_1 = \frac{1}{\sigma_{ts1}} - \frac{1}{\sigma_{cs1}}, f_2 = \frac{1}{\sigma_{ts2}} - \frac{1}{\sigma_{cs2}}, f_3 = \frac{1}{\sigma_{ts3}} - \frac{1}{\sigma_{cs3}}$$

all the other coefficients are set to zero.

JENKINS CRITERION

For Jenkins orthotropic criterion, enter nine coefficients corresponding to the tensile strengths σ_{tsi}, compressive strengths σ_{csi}, and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from

$$g(\sigma) = \max\left(\frac{|\sigma_1|}{\sigma_{s1}}, \frac{|\sigma_2|}{\sigma_{s2}}, \frac{|\sigma_3|}{\sigma_{s3}}, \frac{\sigma_{23}}{\sigma_{ss23}}, \frac{\sigma_{13}}{\sigma_{ss13}}, \frac{\sigma_{12}}{\sigma_{ss12}}\right) - 1$$

here, ε_{si} is either the tensile strength or the compressive strength depending whether the stress in the i direction, σ_i, is positive or negative. The absolute value of the shear stress σ_{ij} in the ij-plane is compared to the corresponding shear strength σ_{ssij}.

WADDOUPS CRITERION

The Waddoups orthotropic criterion is similar to the Jenkins criterion, but the failure criterion is given in terms of strains, not strengths. For this criterion, enter nine coefficients corresponding to the ultimate tensile strains ε_{tsi}, ultimate compressive strains ε_{csi}, and ultimate shear strains γ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from

$$g(\varepsilon) = \max\left(\frac{|\varepsilon_1|}{\varepsilon_{s1}}, \frac{|\varepsilon_2|}{\varepsilon_{s2}}, \frac{|\varepsilon_3|}{\varepsilon_{s3}}, \frac{\gamma_{23}}{\gamma_{ss23}}, \frac{\gamma_{13}}{\gamma_{ss13}}, \frac{\gamma_{12}}{\gamma_{ss12}}\right) - 1$$

here, ε_{si} is either the ultimate tensile strain or the ultimate compressive strain depending whether the strain in the i direction, ε_i, is positive or negative. The absolute

value of the shear strain γ_{ij} in the ij-plane is compared to the corresponding ultimate shear strain γ_{ssij}.

AZZI–TSAI–HILL CRITERION

This criterion is derived from the Tsai–Wu theory for two-dimensional plane stress problems. It is available in 2D for the Plate interface, for the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths σ_{tsi}, compressive strengths σ_{csi}, and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from the in-plane stresses

$$g(\sigma) = F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 - 1$$

The Tsai–Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0 \text{, or } F_{11} = \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts2}^2} \quad \text{for } \sigma_{22} > 0 \text{, or } F_{22} = \frac{1}{\sigma_{cs2}^2} \quad \text{for } \sigma_{22} < 0$$

$$F_{12} = -\frac{1}{2}\frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0 \text{, or } F_{12} = -\frac{1}{2}\frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{66} = \frac{1}{\sigma_{ss12}^2}$$

all the other coefficients are set to zero.

NORRIS CRITERION

This criterion is derived from the Tsai–Wu theory for two-dimensional plane stress problems. It is available in 2D for the Plate interface and the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths σ_{tsi}, compressive strengths σ_{csi}, and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from the in-plane stresses

$$g(\sigma) = F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 - 1$$

The Tsai–Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0 \text{, or } F_{11} = \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts2}^2} \quad \text{for } \sigma_{22} > 0 \text{, or } F_{22} = \frac{1}{\sigma_{cs2}^2} \quad \text{for } \sigma_{22} < 0$$

$$F_{66} = \frac{1}{\sigma_{ss12}^2}$$

$$F_{12} = -\frac{1}{2}\sqrt{F_{11}F_{22}}$$

all the other coefficients are set to zero.

VON MISES CRITERION

The von Mises criterion is one of the simplest isotropic criteria to predict yielding in metals and other ductile materials. The failure criterion is computed from the isotropic tensile strength σ_{ts}

$$g(\sigma) = \frac{\sigma_{mises}}{\sigma_{ts}} - 1$$

The equivalent von Mises stress σ_{mises} is defined from the deviatoric stress tensor, see the section about plasticity and von Mises Criterion. For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

TRESCA CRITERION

Tresca criterion is similar to the von Mises criterion. For this isotropic criterion, the failure criterion is computed from the isotropic tensile strength σ_{ts}

$$g(\sigma) = \frac{\sigma_{tresca}}{\sigma_{ts}} - 1$$

Here, the Tresca equivalent stress is defined in terms of principal stresses, $\sigma_{tresca} = \sigma_1 - \sigma_3$; see Tresca Criterion. For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

RANKINE CRITERION

The Rankine criterion is similar to the Tresca criterion, as the failure criterion is given in terms of principal stresses. For this isotropic criterion, enter the tensile strength σ_{ts}, and the compressive strength σ_{cs}. The failure criterion is then computed from

$$g(\sigma) = \max\left(\frac{|\sigma_{p1}|}{\sigma_s}, \frac{|\sigma_{p2}|}{\sigma_s}, \frac{|\sigma_{p3}|}{\sigma_s}\right) - 1$$

where σ_s is either the tensile strength or the compressive strength depending whether the principal stress, σ_{pi}, is positive or negative. For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

SAINT-VENANT CRITERION

The Saint-Venant criterion is similar to the Waddoups criterion, as the failure criterion is given in terms of strains, not strengths. For this isotropic criterion, enter the ultimate tensile strains, ε_{ts}, and the ultimate compressive strains, ε_{cs}. The failure criterion is then computed from

$$g(\varepsilon) = \max\left(\frac{\varepsilon_{p1}}{\varepsilon_s}, \frac{\varepsilon_{p2}}{\varepsilon_s}, \frac{\varepsilon_{p3}}{\varepsilon_s}\right) - 1$$

Here, ε_s is either the ultimate tensile strain or the ultimate compressive strain depending on whether the principal strain, ε_{pi}, is positive or negative. For ductile materials the ultimate tensile strain corresponds to the strain at yielding, while for brittle materials it corresponds to the strain at failure.

MOHR–COULOMB CRITERION

The Mohr–Coulomb criterion is similar to the Tresca criterion, as the failure criterion is given in terms of principal stresses, see Mohr–Coulomb Criterion for soil plasticity. For this isotropic criterion, enter the cohesion $\mathfrak{c}$, and the angle of internal friction ϕ. The failure criterion is then computed from

$$g(\sigma) = \frac{\sqrt{J_2}m(\theta)}{k - \alpha I_1} - 1$$

and the failure index from

$$fi = \frac{\sqrt{J_2}m(\theta)}{k - \alpha I_1}$$

where

$$m(\theta) = \sqrt{\frac{1}{3}}\left((1+\sin\phi)\cos\theta - (1-\sin\phi)\cos\left(\theta+\frac{2\pi}{3}\right)\right),\ \alpha = \sin\phi/3\text{, and}$$

$$k = c\cos\phi$$

The cohesion and the angle of internal friction are related to the tensile and compressive strengths by the expressions

$$\sin(\phi) = \frac{\sigma_{cs1}-\sigma_{ts1}}{\sigma_{cs1}+\sigma_{ts1}} \text{ and } c\cos(\phi) = \frac{\sigma_{cs1}\sigma_{ts1}}{\sigma_{cs1}+\sigma_{ts1}}$$

DRUCKER–PRAGER CRITERION

The Drucker–Prager criterion approximates the Mohr–Coulomb criterion by a smooth function (a cone in the stress space), see Drucker–Prager Criterion for soil plasticity. The failure isotropic criterion is computed from the stress invariants I_1 and J_2, and two material parameters, α and k,

$$g(\sigma) = \frac{\sqrt{J_2}}{k-\alpha I_1} - 1$$

The material parameters α and k are related to the cohesion c and angle of internal friction ϕ in the Mohr–Coulomb criterion, see Drucker–Prager Criterion for details. Also, the cohesion and the angle of internal friction can be related to the tensile and compressive strengths, see Mohr–Coulomb Criterion for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{k-\alpha I_1}$$

GRIFFITH CRITERION

Griffith criterion (Ref. 111) is intended to study the fracture of brittle materials.

The convex failure surface is given in terms of the principal stresses, the compressive strength, σ_{cs}, and the tensile strength, σ_{ts}

$$g(\sigma) = \frac{\sigma_1-\sigma_3}{(1-m)^2\sigma_{ts}} + \frac{2}{(1-m)}\sqrt{1-\frac{\sigma_1}{\sigma_{ts}}} - 1$$

where the parameter m is computed from the ratio

$$m = \sqrt{1 + \frac{\sigma_{cs}}{\sigma_{ts}}}$$

Griffith's criterion is a paraboloids that opens into the compressive region in the stress space. Authors in Ref. 111 combined it with a Rankine Criterion to avoid tensile stresses higher than the tensile strength σ_{ts}, see Combined Failure Criterion.

BRESLER–PISTER CRITERION

The Bresler–Pister criterion was originally devised to predict the strength of concrete under multiaxial stresses. This isotropic failure criterion is an extension of Drucker–Prager Criterion to brittle materials. The failure criterion is computed from the stress invariants I_1 and J_2, and three parameters, k_1, k_2, and k_3,

$$g(\sigma) = \frac{\sqrt{J_2}}{-k_1 I_1^2 - k_2 I_1 - k_3} - 1$$

The parameters k_1, k_2, and k_3 are computed from the uniaxial compressive strength σ_{cs}, the uniaxial tensile strength σ_{ts}, and the biaxial compressive strength σ_{bc}, see Bresler–Pister Criterion for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{-k_1 I_1^2 - k_2 I_1 - k_3}$$

WILLAM–WARNKE CRITERION

The Willam–Warnke isotropic criterion is used to predict failure in concrete and other cohesive-frictional materials such as rock, soil, and concrete. Just as Bresler–Pister Criterion, failure is computed from the stress invariants I_1 and J_2, and the Lode angle θ, and three material parameters

$$g(\sigma) = \frac{\sqrt{J_2}}{\sqrt{\frac{5}{2}} r(\theta) \sigma_{cs} \left(1 + \left(\frac{1}{\sigma_{bc}} - \frac{1}{\sigma_{ts}}\right)\frac{I_1}{3}\right)} - 1$$

here, σ_{cs} is the compressive strength, σ_{ts} is tensile strength, and σ_{bc} is the biaxial compressive strength. The function $r(\theta)$ describes the segment of an ellipse on the octahedral plane; see Willam–Warnke Criterion for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{\sqrt{\frac{5}{2}} r(\theta)\sigma_{cs}\left(1 + \left(\frac{1}{\sigma_{bc}} - \frac{1}{\sigma_{ts}}\right)\frac{I_1}{3}\right)}$$

OTTOSEN CRITERION

The Ottosen criterion is a five-parameter failure isotropic criterion developed to model short-time loading of concrete. It corresponds to a smooth convex failure surface with curved meridians

$$g(\sigma) = \frac{aJ_2}{\sigma_{cs}(\sigma_{cs} - bI_1)} + \frac{\lambda(\theta)\sqrt{J_2}}{\sigma_{cs} - bI_1} - 1$$

In this formulation, the parameters a and b are positive and dimensionless, and σ_{cs} is the compressive strength for concrete (also with a positive sign). The dimensionless function $\lambda(\theta)$ depends on the Lode angle θ and two positive parameters k_1 and k_2; see Ottosen Criterion for details. The failure index is computed from

$$fi = \frac{aJ_2}{\sigma_{cs}(\sigma_{cs} - bI_1)} + \frac{\lambda(\theta)\sqrt{J_2}}{\sigma_{cs} - bI_1}$$

USER DEFINED

This option makes it possible to explicitly write how the failure criterion and the safety factor depend on stress or strain. These could be analytic functions of stress or strain tensor components, principal stresses, principal strains, stress or strain invariants, or data interpolated from tables.

To replicate the **von Mises Isotropic** criterion with a tensile strength of 350 MPa, define $g(S)$ as `solid.mises/350[MPa]-1` and $s_f(S)$ as `350[MPa]/(solid.mises+eps)`.

COMBINED FAILURE CRITERION

The **Safety** subnode calculates the failure index, damage index, safety factor, and margin of safety for a specified criterion. When multiple **Safety** subnodes are added, additional built-in variables become available. The *combined safety factor*, *combined failure index*, *combined damage index*, and *combined margin of safety*, offer a conservative approach to safety assessment for multiple failure criteria.

The *combined failure criterion* is estimated by selecting the most critical failure criterion for a given multiaxial stress state

$$g_{\text{com}}(\sigma) = \max(g_1, g_2, g_3, \ldots)$$

The *combined failure index* is computed accordingly

$$fi_{\text{com}} = \max(fi_1, fi_2, fi_3, \ldots)$$

and failure is predicted when the combined failure index is greater than one, $fi_{\text{com}} \geq 1$.

The *combined damage index* is given by a Boolean expression based on the combined failure index

$$di_{\text{com}} = \begin{cases} 1 & fi_{\text{com}} \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

where $di_{\text{com}} = 1$ represents a damaged material, and $di_{\text{com}} = 0$ represents a healthy material.

The *combined safety factor* is obtained from the smallest safety factor among the added criteria

$$s_{\text{f,com}} = \min(s_{\text{f1}}, s_{\text{f2}}, s_{\text{f3}}, \ldots)$$

which is a conservative approach.

The *combined margin of safety* is then computed from the combined safety factor $s_{\text{f,com}}$ as

$$ms_{\text{com}} = s_{\text{f,com}} - 1$$

The internal variables fi_{com}, di_{com}, $s_{\text{f,com}}$, and ms_{com} are named *item*`.f_i_com`, *item*`.d_i_com`, *item*`.s_f_com`, and *item*`.m_s_com`, respectively. Here, *item* is the name of the parent node (for instance, `solid.lemm1` for a Linear Elastic material).

FIBER COMPOSITE FAILURE CRITERIA

There are specific criteria for composites that account for the multiaxial stress state and distinct failure modes under tensile and compressive loading. In fiber-reinforced composites, for instance, failure can occur due to fiber rupture under tension or fiber buckling under compression, as well as matrix failure in either tension or compression, among other mechanisms.

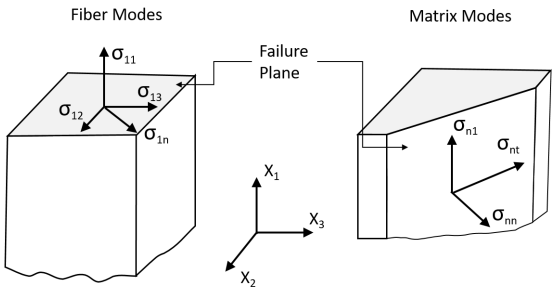


Figure 3-36: The fiber and matrix failure modes and respective failure planes in a unidirectional composite.

The basic types of failure modes in unidirectional composites are (Figure 3-37)

- Fiber rupture in tension
- Fiber buckling in compression
- Matrix failure in tension
- Matrix failure in compression
- Matrix failure for out-of-plane shear
- Matrix failure for in-plane shear
- Interlaminar failure in tension
- Interlaminar failure in compression

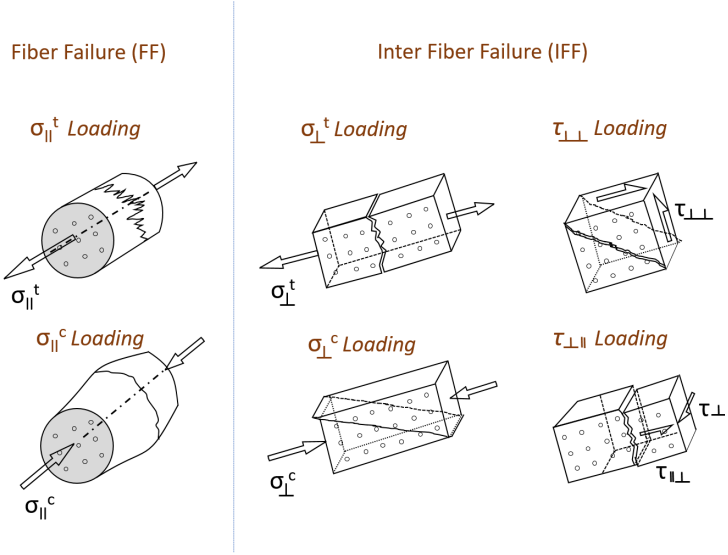


Figure 3-37: Different fiber failure (FF) modes and interfiber failure (IFF) modes of a unidirectional composite under different loading conditions. The plane where the brittle fracture occurs is also shown.

The list of fiber composite failure criteria includes:

- Zinoviev Criterion
- Hashin–Rotem Criterion
- Hashin Criterion
- Puck Criterion
- LARC-03 Criterion

Table 3-8 shows failure criterion and considered failure modes by that criterion.

TABLE 3-8: COMPOSITE FAILURE CRITERIA AND FAILURE MODES.

FAILURE CRITERION	CONSIDERED FAILURE MODE
Zenoviev	Fiber rupture in tension
	Fiber buckling in compression
	Matrix failure in tension
	Matrix failure in compression
	Matrix failure in-plane shear
Hashin–Rotem	Fiber rupture in tension
	Fiber buckling in compression
	Matrix failure in tension and in-plane shear
	Matrix failure in compression and in-plane shear
Hashin	Fiber rupture in tension and shear
	Fiber buckling in compression
	Matrix failure in tension and shear
	Matrix failure in compression and shear
	Interlaminar failure in tension
	Interlaminar failure in compression
Puck	Fiber rupture in tension
	Fiber buckling in compression
	Matrix failure in tension and in-plane shear (Interfiber failure Mode A)
	Matrix failure in compression and in-plane shear (Interfiber failure Mode B)
	Matrix failure in compression and in-plane shear (Interfiber failure Mode C)
LaRC03	Fiber rupture in tension (Mode LaRC03-3)
	Fiber buckling in compression with matrix compression (Mode LaRC03-5)
	Fiber buckling in compression with matrix tension (Mode LaRC03-4)
	Matrix failure in compression and in-plane shear (Mode LaRC03-1)
	Matrix failure in tension and in-plane shear (Mode LaRC03-2)
	Matrix failure in biaxial compression (Mode LaRC03-6)

ZINOVIEV CRITERION

This criterion was developed from a maximum stress failure theory to a well-structured set of noninteracting criteria to identify failure modes. The theory, in general, gives reasonably good failure envelopes for unidirectional laminates and a good fit to the experimental final failure envelopes for multidirectional laminates (Ref. 114). The

approach is one of the most used due to its simplicity. The failure criteria for different failure modes are defined as follows.

The longitudinal failure criterion in tension is

$$g_{1T}(\sigma) = \frac{\sigma_{11}}{\sigma_{ts1}} - 1 \text{ for } \sigma_{11} \geq 0$$

The longitudinal failure criterion in compression is

$$g_{1C}(\sigma) = \left|\frac{\sigma_{11}}{\sigma_{cs1}}\right| - 1 \text{ for } \sigma_{11} < 0$$

The transverse failure criterion in tension is

$$g_{tT}(\sigma) = \frac{\sigma_{22}}{\sigma_{ts2}} - 1 \text{ for } \sigma_{22} \geq 0$$

The transverse failure criterion in compression is

$$g_{tC}(\sigma) = \left|\frac{\sigma_{22}}{\sigma_{cs2}}\right| - 1 \text{ for } \sigma_{22} < 0$$

The in-plane shear failure criterion is

$$g_{s}(\sigma) = \left|\frac{\sigma_{12}}{\sigma_{ss12}}\right| - 1$$

The failure criterion for the composite is then computed by selecting the most critical failure mode

$$g(\sigma) = \max(g_{1T}, g_{1C}, g_{tT}, g_{tC}, g_{s})$$

HASHIN–ROTEM CRITERION

This criterion is based on the first ply failure theory and designed for unidirectional laminates. The Hashin–Rotem failure theory considers tensile and compressive stresses while predicting the failure of fiber or matrix, however it does not distinguish whether the failure occurs at the fiber-matrix interface or inside the matrix. The criterion involves four failure modes associated with fiber failure and the matrix failure, distinguishing between tension and compression.

The fiber failure criterion in tension is defined by

$$g_{\mathrm{fT}}(\sigma) = \frac{\sigma_{11}}{\sigma_{\mathrm{ts1}}} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criterion in compression is defined by

$$g_{\mathrm{fC}}(\sigma) = \left|\frac{\sigma_{11}}{\sigma_{cs1}}\right| - 1 \text{ for } \sigma_{11} < 0$$

The matrix failure criterion in tension is defined by

$$g_{\mathrm{mT}}(\sigma) = \left(\frac{\sigma_{22}}{\sigma_{\mathrm{ts2}}}\right)^2 + \left(\frac{\sigma_{12}}{\sigma_{ss12}}\right)^2 - 1 \text{ for } \sigma_{22} \geq 0$$

The matrix failure criterion in compression is defined by

$$g_{\mathrm{mC}}(\sigma) = \left(\frac{\sigma_{22}}{\sigma_{cs2}}\right)^2 + \left(\frac{\sigma_{12}}{\sigma_{ss12}}\right)^2 - 1 \text{ for } \sigma_{22} < 0$$

The failure criterion for the composite is then computed by selecting the most critical failure mode

$$g(\sigma) = \max(g_{\mathrm{fT}}, g_{\mathrm{fC}}, g_{\mathrm{mT}}, g_{\mathrm{mC}})$$

HASHIN CRITERION

The Hashin failure theory is an extension of Hashin–Rotem theory, as it includes six failure modes for fiber failure, matrix failure and interlaminar failure (Ref. 113). Stress interactions are considered for the determination of the fiber tensile failure mode, matrix tensile failure mode, and matrix compressive failure mode.

The fiber failure criterion in tension is defined by

$$g_{\mathrm{fT}}(\sigma) = \frac{\sigma_{11}^2}{\sigma_{\mathrm{ts1}}^2} + \frac{\sigma_{12}^2 + \sigma_{13}^2}{\sigma_{\mathrm{ss12}}^2} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criterion in compression is defined by

$$g_{\mathrm{fC}}(\sigma) = \left|\frac{\sigma_{11}}{\sigma_{cs1}}\right| - 1 \text{ for } \sigma_{11} < 0$$

The matrix failure criterion in tension is defined by

$$g_{mT}(\sigma) = \frac{(\sigma_{22}+\sigma_{33})^2}{\sigma_{ts2}^2} + \frac{\sigma_{23}^2 - \sigma_{22}\sigma_{33}}{\sigma_{ss23}^2} + \frac{\sigma_{12}^2+\sigma_{13}^2}{\sigma_{ss12}^2} - 1 \text{ for } \sigma_{22}+\sigma_{33} \geq 0$$

The matrix failure criterion in compression is defined by

$$g_{mC}(\sigma) = \left(\frac{\sigma_{cs2}^2}{4\sigma_{ss23}^2} - 1\right)\frac{\sigma_{22}+\sigma_{33}}{\sigma_{cs2}} + \frac{(\sigma_{22}+\sigma_{33})^2}{4\sigma_{ss23}^2} + \frac{\sigma_{23}^2 - \sigma_{22}\sigma_{33}}{\sigma_{ss23}^2} + \frac{\sigma_{12}^2+\sigma_{13}^2}{\sigma_{ss12}^2} - 1$$

$$\text{for } \sigma_{22}+\sigma_{33} < 0$$

The interlaminar failure criterion in tension is defined by

$$g_{iT}(\sigma) = \frac{\sigma_{33}^2}{\sigma_{ts3}^2} - 1 \text{ for } \sigma_{33} \geq 0$$

The interlaminar failure criterion in compression is defined by

$$g_{iC}(\sigma) = \frac{\sigma_{33}^2}{\sigma_{cs3}^2} - 1 \text{ for } \sigma_{33} < 0$$

The failure criterion for the composite is then computed by selecting the most critical failure mode

$$g(\sigma) = \max(g_{fT}, g_{fC}, g_{mT}, g_{mC}, g_{iT}, g_{iC})$$

The plane stress version of Hashin criterion is obtained by setting $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$, however, note that interlaminar failure cannot then be predicted.

PUCK CRITERION

The Puck criterion is based on 3D phenomenological models, where the experimental results are matched with theoretical formulation. Based on fracture mechanics and experimental observation, three different failure criteria called as modes A, B, and C for matrix failure are considered. Mode A corresponds to tensile loading, while mode B and mode C correspond to compressive and shear loading.

The Puck criterion is used for predicting strength of unidirectional laminate, and for predicting the initial strength of multidirectional laminates for which the other criteria do not predict failure correctly (Ref. 114). The criterion treats separately fiber failure (FF) and interfiber failure (IFF).

Fiber failure in tension is defined from the expression

$$g_{\mathrm{fT}}(\sigma,\varepsilon) = \frac{1}{\varepsilon_{\mathrm{ts1}}}\left(\varepsilon_{11} + \frac{\upsilon_{\mathrm{f12}}}{E_{\mathrm{f1}}} m_{\sigma\mathrm{f}}\sigma_{22}\right) - 1 \text{ for } \varepsilon_{11} \geq 0$$

where E_{f1} is the Young's modulus of the fiber, ν_{f12} is its in-plane Poisson's ratio, $m_{\sigma\mathrm{f}}$ is the *mean stress magnification factor*, $\varepsilon_{\mathrm{ts1}}$ is the ultimate tensile strain of the fiber, and ε_{11} is the strain in the longitudinal direction.

Fiber failure in compression is defined from the expression

$$g_{\mathrm{fC}}(\sigma,\varepsilon) = \frac{1}{\varepsilon_{\mathrm{cs1}}}\left|\left(\varepsilon_{11} + \frac{\upsilon_{\mathrm{f12}}}{E_{\mathrm{f1}}} m_{\sigma\mathrm{f}}\sigma_{22}\right)\right| + (10\gamma_{12})^2 - 1 \text{ for } \varepsilon_{11} < 0$$

where $\varepsilon_{\mathrm{cs1}}$ is the ultimate compressive strain of the fiber, and γ_{12} is its in-plane shear strain.

The failure criterion for interfiber failure mode A is

$$g_{\mathrm{mA}}(\sigma) = \sqrt{\left(\frac{\sigma_{12}}{\sigma_{\mathrm{ss12}}}\right)^2 + \left(1 - \frac{\sigma_{\mathrm{ts2}} p_{\mathrm{tl}}}{\sigma_{\mathrm{ss12}}}\right)^2 \left(\frac{\sigma_{22}}{\sigma_{\mathrm{ts2}}}\right)^2} + \frac{\sigma_{22} p_{\mathrm{tl}}}{\sigma_{\mathrm{ss12}}} + \left|\frac{\sigma_{11}}{\sigma_{\mathrm{1D}}}\right| - 1 \text{ for } \sigma_{22} \geq 0$$

where p_{tl} is the *slope of the in-plane fracture envelope in tension*, and σ_{1D} is the *linear degradation stress.*

The failure criterion for interfiber failure mode B is

$$g_{\mathrm{mB}}(\sigma) = \frac{1}{\sigma_{\mathrm{ss12}}}\left(\sqrt{\sigma_{12}^2 + (\sigma_{22} p_{\mathrm{cl}})^2} + \sigma_{22} p_{\mathrm{cl}}\right) + \left|\frac{\sigma_{11}}{\sigma_{\mathrm{1D}}}\right| - 1$$

$$\text{for } \sigma_{22} < 0 \text{ and } 0 \leq \left|\frac{\sigma_{22}}{\sigma_{12}}\right| \leq \left|\frac{R^{\mathrm{A}}_{\mathrm{tt}}}{\sigma^{\mathrm{c}}_{\mathrm{ss12}}}\right|$$

where p_{cl} is the *slope of the in-plane fracture envelope in compression*, $R^{\mathrm{A}}{}_{\mathrm{tt}}$ is the *fracture resistance against transverse shear loading*, and $\sigma^{\mathrm{c}}{}_{\mathrm{ss12}}$ is the modified shear strength.

The failure criterion for interfiber failure mode C is

$$g_{\mathrm{mC}}(\sigma) = \frac{\sigma_{\mathrm{cs2}}}{-\sigma_{22}}\left(\left(\frac{\sigma_{12}}{2(1 + p_{\mathrm{ct}})\sigma_{\mathrm{ss12}}}\right)^2 + \left(\frac{\sigma_{22}}{\sigma_{\mathrm{cs2}}}\right)^2\right) + \left|\frac{\sigma_{11}}{\sigma_{\mathrm{1D}}}\right| - 1$$

$$\text{for } \sigma_{22} < 0 \text{ and } 0 \le \left|\frac{\sigma_{12}}{\sigma_{22}}\right| \le \left|\frac{\sigma^{c}_{ss12}}{R^{A}_{tt}}\right|$$

where p_{ct} is the *slope of the transverse fracture envelope in compression*.

The failure criterion for the composite is then computed by selecting the most critical failure mode

$$g(\sigma) = \max(g_{fT}, g_{fC}, g_{mA}, g_{mB}, g_{mC})$$

LARC-03 CRITERION

This criterion is used for accurate predicting the failure of unidirectional FRP laminates with in-plane stress state. The criterion is composed of six phenomenological failure modes that describe matrix and fiber failure without the use of curve-fitting parameters (Ref. 115). Furthermore, it assumes a fragile fracture for matrix failure in compression. This criterion implements the *action plane* concept according to the Mohr–Coulomb theory, and it also considers failure modes based on both fiber kinking due to misalignment and on tensile matrix cracking associated with interlaminar crack propagation.

The matrix failure criterion under transverse compression (LaRC03-1) is

$$g_{mC}(\sigma) = \left(\frac{\tau_{eff,t}}{\sigma_{ss23}}\right)^2 + \left(\frac{\tau_{eff,l}}{\sigma^{i}_{ss12}}\right)^2 - 1 \text{ for } \sigma_{11} \ge -\sigma_{cs2} \text{ and } \sigma_{22} < 0$$

where $\tau_{eff,t}$ and $\tau_{eff,l}$ are the effective shear stresses in the transverse and longitudinal directions, respectively, σ^{i}_{ss12} is the *in situ in-plane shear strength*, σ_{ss23} is the shear strength in the transverse direction, and σ_{cs2} is the compressive strength in the transverse direction. The effective shear stresses are functions of the fracture plane angle which is found by maximizing the Mohr–Coulomb effective stresses.

The matrix failure criterion under transverse tension (LaRC03-2) is

$$g_{mT}(\sigma) = (1-r)\left(\frac{\sigma_{22}}{\sigma^{i}_{ts2}}\right) + r\left(\frac{\sigma_{22}}{\sigma^{i}_{ts2}}\right)^2 + \left(\frac{\sigma_{12}}{\sigma^{i}_{ss12}}\right)^2 - 1 \text{ for } \sigma_{22} \ge 0$$

where σ^{i}_{ts2} is the *in situ transverse tensile strength*, and r is a dimensionless parameter based on fracture toughness.

The fiber failure criterion under longitudinal tension (LaRC03-3) is

$$g_{fT}(\varepsilon) = \frac{\varepsilon_{11}}{\varepsilon_{ts1}} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criterion under matrix tension (LaRC03-4) is

$$g_{fCmT}(\sigma) = (1-r)\left(\frac{\sigma^m_{22}}{\sigma^i_{ts2}}\right) + r\left(\frac{\sigma^m_{22}}{\sigma^i_{ts2}}\right)^2 + \left(\frac{\sigma^m_{12}}{\sigma^i_{ss12}}\right)^2 - 1 \text{ for } \sigma_{22} \geq 0$$

The fiber failure criterion under matrix compression (LaRC03-5) is

$$g_{fCmC}(\sigma) = \left\langle \frac{\left|\sigma^m_{12}\right| + \sigma^m_{22}\eta_l}{\sigma^i_{ss12}} \right\rangle - 1 \text{ for } \sigma_{22} < 0$$

where σ^m_{ij} are the ply stresses transformed to the misalignment coordinate frame, and η_l is a nondimensional parameter based on the failure strength and fracture plane angle under uniaxial transverse compression.

The matrix failure criterion under biaxial compression (LaRC03-6) is

$$g_{mB}(\sigma) = \left(\frac{\tau^m_{eff,t}}{\sigma_{ss23}}\right)^2 + \left(\frac{\tau^m_{eff,l}}{\sigma^i_{ss12}}\right)^2 - 1 \text{ for } \sigma_{11} < -\sigma_{cs2} \text{ and } \sigma_{22} < 0$$

where the effective shear stresses in the transverse and longitudinal directions, $\tau^m_{eff,t}$ and $\tau^m_{eff,l}$, are calculated from the stresses in the misaligned coordinates.

The failure criterion for the composite is then computed by selecting the most critical failure mode

$$g(\sigma) = \max(g_{mC}, g_{mT}, g_{fT}, g_{fCmT}, g_{fCmC}, g_{mB})$$

Formulation of the Equilibrium Equations

- Equation Implementation

Equation Implementation

The equilibrium equations for solid mechanics are given by Newton's second law. It is usually written using a spatial formulation in terms of the Cauchy stress tensor σ:

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_x\sigma + \mathbf{f}_V$$

Here $\mathbf{f}_V$ is a body force per unit of deformed volume, and ρ is the current mass density. For the material frame formulation used in COMSOL Multiphysics, it is more appropriate to use a Lagrangian version if the equation:

$$\rho_0\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_X P^T + \mathbf{F}_V$$

Now that the first Piola–Kirchhoff stress tensor, P, is used. $\mathbf{F}_V$ is a body force with components in the current configuration but given with respect to the undeformed volume, and ρ_0 is the initial mass density. Note the gradient operators are not the same: in the first case the gradient is taken with respect to the spatial coordinates, and in the second case with respect to the material coordinates. Using the more common second Piola–Kirchhoff stress tensor, S, the same equation reads

$$\rho_0\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_X(FS) + \mathbf{F}_V \tag{3-189}$$

where F is the deformation gradient. The COMSOL Multiphysics implementation of the equations in the Solid Mechanics interface is however not based on the equation of motion directly, but rather on the *principle of virtual work*.

The principle of virtual work states that the sum of the internal virtual work and the external virtual work are equal. The internal virtual work is the work done by the current stress state on a kinematically admissible variation in strains. The external

virtual work is the work done by all forces (acting on domains, boundaries, edges, or points) when multiplied with the variation in displacements corresponding to the variation in strains. The virtual displacements $\delta\mathbf{u}$ are in the finite element formulation represented by the `test()` operator in COMSOL Multiphysics. For a stationary case, the virtual work δW is written as

$$\delta W = \int_V (-\delta\varepsilon : \sigma + \delta\mathbf{u} \cdot \mathbf{F}_V) dv +$$
$$\int_S (\delta\mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta\mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta\mathbf{u} \cdot \mathbf{F}_p)]$$

The strains are computed from the gradients of the displacements, and the stresses are given by the constitutive relation.

In a dynamic analysis, the inertial forces are included in the volume forces, according to d'Alembert's principle.

$$\delta W = \int_V (-\delta\varepsilon : \sigma + \delta\mathbf{u} \cdot \mathbf{F}_V - \rho\delta\mathbf{u} \cdot \mathbf{u}_{tt}) dv + \quad (3\text{-}190)$$
$$\int_S (\delta\mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta\mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta\mathbf{u} \cdot \mathbf{F}_p)$$

Since the equations are formulated on the material frame, all integrals are taken over the undeformed geometry. The stress and strain contributions must be interpreted differently depending on whether the formulation is geometrically nonlinear or not.

- If the study step is geometrically linear, the strain ε is the engineering strain. The stress σ could in principle be any of the stress measures, as they all converge to the same engineering stress in this case.
- If the study step is geometrically nonlinear, the variation of strain $\delta\varepsilon$ is taken from the displacement gradient, and the stress conjugate σ is the first Piola–Kirchhoff stress. The integration $\int_V dv$ is done in the undeformed configuration.

The Solid Mechanics interface supports Stationary (static), Eigenfrequency, Time Dependent (transient), Frequency Domain, and Modal solver study types as well as *linear buckling*.

Study Types

In this section:

- Frequency-Domain Studies
- Eigenfrequency Studies
- Random Vibration Theory
- Response Spectrum Analysis Theory
- Linear Buckling

Frequency-Domain Studies

In the frequency domain, the frequency response is studied when applying harmonic loads. Harmonic loads are specified using two components:

- The amplitude value in direction m, F_m
- The corresponding phase, ϕ_m

To derive the equations for the linear response from harmonic excitation loads

$$F_{m,\,\text{freq}} = F_m(f)\cos(\omega t + \phi_m)$$

$$\mathbf{F}_{\text{freq}} = \begin{bmatrix} F_{x,\,\text{freq}} \\ F_{y,\,\text{freq}} \\ F_{z,\,\text{freq}} \end{bmatrix}$$

Assume a harmonic response with the same angular frequency as the excitation load

$$u = u_{\text{amp}}\cos(\omega t + \phi_u)$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The relationship can also be described using complex notation with

$$u = \text{Re}(u_{\text{amp}} e^{j\phi_u} e^{j\omega t}) = \text{Re}(\tilde{u} e^{j\omega t}) \text{ where } \tilde{u} = u_{\text{amp}} e^{j\phi_u}$$

$$\mathbf{u} = \mathrm{Re}(\tilde{\mathbf{u}} e^{j\omega t})$$

and

$$F_{m,\,\mathrm{freq}} = \mathrm{Re}(F_m(\omega) e^{j\phi_m} e^{j\omega t}) = \mathrm{Re}(\tilde{F}_m e^{j\omega t})$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{bmatrix}$$

The primary results, such as displacements, velocities, accelerations, and stress and strain components are all complex valued.

CYCLE MAXIMUM

Quantities that are nonlinear with respect to the degrees of freedom will in general not have a harmonic variation. In some cases, it is however possible to derive maximum values for a cycle. The norm of a vector is defined as

$$|\mathbf{v}(t)| = \sqrt{\sum_j v_j(t)^2}$$

The components of the vector can have different phases, so that the temporal variation is

$$v_j(t) = a_j \cos(\omega t + \varphi_j)$$

where a_j is the amplitude, and φ_j is the phase angle. Then,

$$\sum_j v_j(t)^2 = \sum_j a_j{}^2 (\cos(\omega t + \varphi_j))^2 = \sum_j \frac{a_j{}^2}{2} (1 + \cos(2\omega t + 2\varphi_j))$$

The square of the norm is thus a harmonic function with double the excitation frequency. The harmonic term can, using standard trigonometric expressions be rewritten as

$$\frac{1}{2} \sum_j a_j{}^2 \cos(2\omega t + 2\varphi_j) = \frac{1}{2} a \cos(2\omega t + \varphi)$$

where

$$a = \sqrt{\sum_j \sum_k a_j^2 a_k^2 \cos(2\varphi_j - 2\varphi_k)}$$

The maximum value of the vector norm during the cycle is thus

$$|\mathbf{v}(t)|_{\max} = \sqrt{\frac{1}{2}\left(\sum_j a_j^2 + a\right)} = \sqrt{\frac{1}{2}\left(\sum_j a_j^2 + \sqrt{\sum_j \sum_k a_j^2 a_k^2 \cos(2\varphi_j - 2\varphi_k)}\right)}$$

The expressions above can be generalized to any quadratic form of the type

$$\sqrt{\sum_j \sum_k c_{jk} v_j(t) v_k(t)}$$

In particular, the von Mises equivalent stress is of this form, with

$$\mathbf{v} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{bmatrix}$$

$$c = \frac{1}{2}\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix}$$

Using the same development as above,

$$\sum_j \sum_k c_{jk} v_j(t) v_k(t) = \sum_j \sum_k c_{jk} a_j a_k \cos(\omega t + \varphi_j) \cos(\omega t + \varphi_k) =$$
$$\sum_j \sum_k \frac{c_{jk} a_j a_k}{2} [\cos(\varphi_j - \varphi_k) + \cos(2\omega t + \varphi_j + \varphi_k)]$$

This expression can be split into a constant (period average) term, and a periodic term, so that

$$\sum_j \sum_k c_{jk} v_j(t) v_k(t) = \frac{1}{2}(b + a\cos(2\omega t + \varphi))$$

where

$$b = \sum_j \sum_k c_{jk} a_j a_k \cos(\varphi_j - \varphi_k)$$

and

$$a = \sqrt{\sum_j \sum_k \sum_l \sum_m c_{jk} c_{lm} a_j a_k a_l a_m \cos(\varphi_j + \varphi_k - \varphi_l - \varphi_m)}$$

The maximum value of the norm is thus

$$\sqrt{\frac{1}{2}(b + a)} = \sqrt{\sum_j \sum_k c_{jk} a_j a_k \cos(\varphi_j - \varphi_k) + \sqrt{\sum_j \sum_k \sum_l \sum_m c_{jk} c_{lm} a_j a_k a_l a_m \cos(\varphi_j + \varphi_k - \varphi_l - \varphi_m)}}$$

Since, for the von Mises stress, the matrix c is symmetric and have many zero elements, the number of terms in the sums that actually contribute are far less than the nominal number.

When computing the cycle maximum of the von Mises stress, there is an alternative, simpler approach. The von Mises stress can be written as

$$\sigma_{\mathrm{vM}} = \sqrt{\frac{3}{2} s : s}$$

where s is the deviatoric stress defined as

$$s = \sigma - \frac{1}{3}\mathrm{trace}(\sigma)\mathbf{I}$$

Note that on this form, there are no cross products between different components of the tensor. By defining

$$\mathbf{v} = \begin{bmatrix} \sqrt{\frac{3}{2}} s_{xx} \\ \sqrt{\frac{3}{2}} s_{yy} \\ \sqrt{\frac{3}{2}} s_{zz} \\ \sqrt{3} s_{xy} \\ \sqrt{3} s_{xz} \\ \sqrt{3} s_{yz} \end{bmatrix}$$

the expression for the maximum of a vector norm can be used directly to compute the maximum von Mises stress.

Eigenfrequency Studies

The eigenfrequency equations are derived by assuming a harmonic displacement field, similar as for the frequency response formulation. The difference is that this study type uses a new variable $j\omega$ explicitly expressed in the eigenvalue $j\omega = -\lambda$. The eigenfrequency f is then derived from $j\omega$ as

$$f = -\frac{\lambda}{2\pi j}$$

Damped eigenfrequencies can also be studied, so λ is not necessarily a purely imaginary number. Any damping included in the problem will automatically cause the eigenfrequencies to become complex valued.

In addition to the eigenfrequency, the quality factor, Q, and decay factor, δ, for the model can be examined:

$$Q = \frac{\mathrm{Im}(\lambda)}{2\mathrm{Re}(\lambda)}$$

$$\delta = \mathrm{Re}(\lambda)$$

MODAL PARTICIPATION FACTORS

It is common to present modal participation factors in terms of the discretized system of equations, that is on matrix form. For a discretized system, the modal mass for the ith mode can be defined as

$$m_i = \sum_{n,m} \phi_i^n M_{nm} \phi_i^m$$

where M_{nm} is the mass matrix, and ϕ_i is the eigenmode in terms of a vector of degrees of freedom. The modal participation factors are defined as

$$\Gamma_{ij} = \frac{1}{m_i} \sum_{n,m} \phi_i^n M_{nm} \psi_j^m$$

where ψ_j^m represent the unit rigid-body modes for translation and rotation.

The effective mass in direction j for mode i can then be computed as

$$m_{ij}^{\text{eff}} = m_i \Gamma_{ij}^2$$

For physics interfaces with only displacements as dependent variables (for example Solid Mechanics), each eigenmode is given by the solution vector $\mathbf{u} = \begin{bmatrix} u & v & w \end{bmatrix}$. The rigid body modes ψ can be represented as columns of the following matrix:

$$\begin{bmatrix} \psi_{TX} & \psi_{TY} & \psi_{TZ} & \psi_{RX} & \psi_{RY} & \psi_{RZ} \\ L & 0 & 0 & 0 & Z - Z_0 & -(Y - Y_0) \\ 0 & L & 0 & -(Z - Z_0) & 0 & X - X_0 \\ 0 & 0 & L & Y - Y_0 & -(X - X_0) & 0 \end{bmatrix}$$

where L represents a unit length.

The translational and rotational participation factors can be computed as, respectively:

$$\Gamma_T = \begin{bmatrix} \Gamma_{TX} & \Gamma_{TY} & \Gamma_{TZ} \end{bmatrix} = \frac{L}{m_F} \int \mathbf{u} dm$$

and

$$\Gamma_R = \begin{bmatrix} \Gamma_{RX} & \Gamma_{RY} & \Gamma_{RZ} \end{bmatrix} = \frac{1}{m_F} \int (\mathbf{r} - \mathbf{r}_0) \times \mathbf{u} dm$$

where the normalization factor is computed as

$$m_F = \int (\mathbf{u}^T \cdot \mathbf{u}) dm$$

The integration involves the entire selection of the corresponding physics interface. The definition of dm in the above formulas depends on the dimensions. For example, one has $dm = \rho dV$ for solid domains in 3D. Contributions to the structural mass come not only from the mass density of the domains, but also from features like **Rigid Connector**, **Added Mass**, **Point Mass**, and so on. Thus, integrations are in general performed over all selected domains, boundaries, and edges. Contributions from points are also added.

If the mass matrix normalization was selected when computing the eigenmodes, then $m_F = 1$.

Note that the rotational participation factor computed with respect to a certain reference point $\mathbf{r}_0$ can be expressed in terms of the participation factors computed with respect to the origin as

$$\Gamma_R(\mathbf{r}_0) = \Gamma_R(0) + \Gamma_T \times \mathbf{r}_0$$

For structural elements (and features) that also use rotational degrees of freedom as dependent variables, there is also a direct contribution from these degrees of freedom. In this case, the corresponding expressions are:

$$\Gamma_R = \frac{1}{m_F}\int(\mathbf{r} \times \mathbf{u} + J\theta)dm$$

with

$$m_F = \int(\mathbf{u}^T \cdot \mathbf{u} + \theta^T J\theta)dm$$

where matrix J presents the moment of inertia, and $\theta = \left[\theta_X\ \theta_Y\ \theta_Z\right]$ are rotation angles with respect to the corresponding axes. The angles can be computed at given local position as certain functions of the actual rotational degrees of freedom which can be different for different structural element types.

An alternative definition of the participation factors is

$$\Gamma^{\text{norm}} = \Gamma\sqrt{\frac{m_F}{ML^2}}$$

where M denotes a unit mass. The advantage of such definition is independence of the normalization type selected when computing the eigenmodes.

The effective modal mass for X-translation and rotation are defined, respectively, as

$$m_{TX}^{\text{eff}} = \frac{m_F}{L^2}\Gamma_{TX}^2 = M(\Gamma_{TX}^{\text{norm}})^2 \text{ and } m_{RX}^{\text{eff}} = m_F\Gamma_{RX}^2$$

Similar definitions are used for other components.

	Only structural mechanics physics interfaces contribute to the mass and participation factor variables as described in this section. In a multiphysics context, there can be also other contributions to the total mass matrix. This would for example be the case when coupling acoustics and structural mechanics. Such contributions are not taken into account.

Random Vibration Theory

Random vibration analysis is applicable in a situation where the loading has a random variation which can be statistically described by its power spectral density (PSD). The duration of the loading must be such that the situation can be considered as a steady state. The key assumptions are:

- The system is linear and can be represented by a superposition of a limited number if eigenmodes.
- The process is ergodic, meaning that its statistical properties do not change over time.

STATISTICS PRELIMINARIES

Assume a random process $x(t)$. In the following, T denotes a time span which is 'long' compared to any frequencies of interest.

	The meaning of a "long" time T in this context is that any vibration mode will experience a large number of cycles. If the lowest frequency of interest is f_1, then its corresponding period is $T_1 = 1/f_1$. It is then reasonable to require that $T > kT_1$, where k is at least 1000, but probably more.

The *mean value* is defined using the expectancy value operator $E[\,]$ as

$$E[x] = \lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)dt = m$$

In vibration problems most quantities have zero mean values. Nonlinear quantities, like an equivalent stress, will however have nonzero mean values. Also, if the total effect of a static preload and the vibration is considered, the total result will in general have a nonzero mean value.

The variance (the square of the standard deviation) is defined as

$$E[(x-m)^2] = E[x^2] - m^2$$

The standard deviation is in this context usually called the root mean square (RMS) value of the signal, x_{rms}.

A representation of the "degree of similarity over time" of a signal is given by the autocorrelation function

$$R_x(\tau) = E[x(t)x(t+\tau)] = \lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)x(t+\tau)dt$$

It can be seen from the definition that the autocorrelation is an even function of the time difference τ. An interpretation of the autocorrelation is that it compares how similar the signal is to itself after a certain time has elapsed.

Also, it can be seen from the definitions above that

$$x_{\text{rms}}^2 = R_x(0)$$

For two different processes $x(t)$ and $y(t)$, the cross-correlation is similarly defined as

$$R_{xy}(\tau) = E[x(t)y(t+\tau)] = \lim_{T\to\infty}\frac{1}{T}\int_0^T x(t)y(t+\tau)dt$$

The power spectral density (PSD) of a signal x is defined through the Fourier transform (denoted by $\mathrm{F}[\,]$) of the autocorrelation,

$$G_x(f) = 2\mathrm{F}[R_x(\tau)] = 2\int_{-\infty}^{\infty} R_x(\tau)e^{-2\pi i f\tau}d\tau$$

The term PSD comes from the fact that the physical dimension of G_x is (input signal)^2 / frequency. Thus, $G_x(f)\Delta f$ represents the power of the signal contained in the small frequency interval Δf.

The inverse relation is

$$R_x(\tau) = \frac{1}{2}\mathrm{F}^{-1}[G_x(f)] = \frac{1}{2}\int_{-\infty}^{\infty} G_x(f)e^{2\pi i f\tau}df$$

Since the autocorrelation is a real-valued and even function of τ, G_x is also real-valued. Because of this property, G_x can also be written as

$$G_x(f) = 4\int_0^{\infty} R_x(\tau)\cos(2\pi f\tau)d\tau$$

The RMS value can be derived from the PSD by integrating the PSD over all frequencies:

$$x_{\mathrm{rms}}^2 = R_x(0) = \frac{1}{2}\int_{-\infty}^{\infty} G_x(f)df = \int_0^{\infty} G_x(f)df$$

It is possible to compute the mean square of x (the power) for only a certain frequency range

$$E[x^2](f_1,f_2) = \int_{f_1}^{f_2} G_x(f)df$$

This can be interpreted as the energy content of the signal between the two frequencies.

In the same way as for the PSD, the cross spectral density for two signals is defined as

$$G_{xy}(f) = 2\mathrm{F}[R_{xy}(\tau)]$$

The cross spectral density is in general a complex-valued function, and G_{yx} is the complex conjugate of G_{xy}.

It is often more convenient to work with the angular frequency $\Omega = 2\pi f$, and define the PSD as

$$S_x(\Omega) = \frac{1}{2\pi}G_x(f)$$

so that

$$S_x = \frac{2}{\pi}\int_0^\infty R_x(\tau)\cos(\Omega\tau)d\tau$$

and

$$x_{\text{rms}}^2 = \int_0^\infty S_x(\Omega)d\Omega$$

There are some other scalar statistical properties that can be used to characterize the properties of the spectrum. Define the k^{th} moment of the spectrum as

$$m_k = \int_0^\infty f^k G(f)df$$

The moments can be used to compute the following properties:

- RMS

$$x_{\text{rms}} = \sqrt{m_0}$$

- Number of (upward) zero crossings per unit time, also called *apparent frequency*

$$n_0 = \sqrt{\frac{m_2}{m_0}}$$

- Number of peaks per unit time

$$n_{\text{p}} = \sqrt{\frac{m_4}{m_2}}$$

- Irregularity factor

$$\gamma = \frac{n_0}{n_{\text{p}}}$$

TRANSFER FUNCTION

For a linear system, the response in the frequency domain for a single variable u to the input x can be written

$$u(\Omega) = H(\Omega)x(\Omega)$$

where H is the complex-valued transfer function. It can then be shown that the corresponding spectral densities have the relation

$$S_u(\Omega) = \left|H_{ux}(\Omega)\right|^2 S_x(\Omega) = H^*{}_{ux}(\Omega) H_{ux}(\Omega) S_x(\Omega)$$

Here, the superscript '*' denotes complex conjugate. When generalizing to matrices, it denotes the Hermitian conjugate (that is transpose and conjugation).

This type of relation is true not only for the degrees of freedom, but for any quantity that is linearly related to the input. This includes components of stress and engineering strain, but not quantities as equivalent or principal stresses.

When there are multiple input signals (loads), the situation is more complicated. If the inputs are uncorrelated (all cross correlations between them are zero), then the resulting output spectrum can be obtained as a pure superposition of the input spectra weighted by the transfer functions

$$S_u = \sum_k \left|H_{ux_k}(\Omega)\right|^2 S_{x_k}$$

When the input signals are correlated, the loading must be described by a complete matrix of cross correlation spectra, where the diagonal consists of the individual PSD. Placing the transfer functions from all inputs (N) to all outputs (M) into a rectangular matrix $\mathbf{H}$ of size M-by-N, the operation of computing the output spectral densities and cross correlation spectra can formally be written as

$$\mathbf{S}_U = \mathbf{H}(\Omega)\mathbf{S}_X(\Omega)\mathbf{H}^*(\Omega)$$

$\mathbf{S}_X$ is a square matrix having the size of number of inputs, N, while the size of the square matrix $\mathbf{S}_U$ is the number of outputs, M.

In practice, large sets of output cross-correlation data are uncommon. The most common cases are:

- Only the diagonal of $\mathbf{S}_U$ is important. It represents the response of a certain quantity without information about correlation to other quantities. If you are designing a structure, you are interested in the size of the stress, not its covariation with the stress somewhere else.
- A small set of full correlations are sought. This is the case where the current analysis provides input for another analysis, so that a full spectral input is needed for a number of attachment points.

MODAL REPRESENTATION

The random vibration analysis is implemented based on a mode superposition, encapsulated in a reduced-order model. This approach imposes some limitations:

- The material data cannot be frequency dependent.
- There are no inhomogeneous Dirichlet conditions. The common case when there is a fully correlated base excitation can however be handled by shifting the base acceleration into an equivalent inertial load. The **Base Excitation** feature is used for this.

In the modal representation, the assumption that the displacements can be described as

$$\mathbf{u}(t) = \sum_k \mathbf{y}_k q_k(t) = \mathbf{Y}\mathbf{q}(t)$$

where the vector $\mathbf{y}_k$ contains eigenmode k, and the mode matrix $\mathbf{Y}$ has the eigenmodes $\mathbf{y}_k$ as columns. The modal coordinates (amplitudes of each mode) are collected in the vector $\mathbf{q}$.

Using the common notation that $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix and $\mathbf{K}$ is the stiffness matrix, the full set equations of motion is

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{L}$$

Projecting to the modal base, the corresponding equations are

$$\mathbf{m}\ddot{\mathbf{q}} + \mathbf{c}\dot{\mathbf{q}} + \mathbf{k}\mathbf{q} = \mathbf{l}$$

Lowercase matrices are used to indicate a quantity in the modal space.

The projections for the matrices are

$$\mathbf{m} = \mathbf{Y}^T\mathbf{M}\mathbf{Y}$$
$$\mathbf{c} = \mathbf{Y}^T\mathbf{C}\mathbf{Y}$$
$$\mathbf{k} = \mathbf{Y}^T\mathbf{K}\mathbf{Y}$$
$$\mathbf{l} = \mathbf{Y}^T\mathbf{L}$$

In frequency domain, any quantity a can be written using a complex notation as

$$a = \hat{a}e^{i\Omega t}$$

where $\hat{a}$ is a complex-valued amplitude. With an assumption about harmonic excitation,

$$(-\Omega^2\mathbf{m} + i\Omega\mathbf{c} + \mathbf{k})\hat{\mathbf{q}} = \hat{\mathbf{l}}$$

In terms of a transfer function, this can be written as

$$\hat{\mathbf{q}} = \mathbf{h}(\Omega)\hat{\mathbf{l}}$$

where

$$\mathbf{h}(\Omega) = (-\Omega^2\mathbf{m} + i\Omega\mathbf{c} + \mathbf{k})^{-1}$$

If the applied load is given in terms of its cross-correlation spectra, $\mathbf{S}_L(\Omega)$, then it can be shown that the corresponding cross correlation for the modal loads, $\mathbf{S}_l(\Omega)$, is obtained by a similar projection into modal space,

$$\mathbf{s}_l(\Omega) = \mathbf{Y}^T\mathbf{S}_L(\Omega)\mathbf{Y}$$

Since the number of modes used typically is rather small, the projected cross-correlation matrix is of a manageable size.

It is now possible to compute the cross-correlation spectrum for the modal degrees of freedom as

$$\mathbf{s}_q(\Omega) = \mathbf{h}(\Omega)\mathbf{s}_l(\Omega)\mathbf{h}^*(\Omega)$$

The cross-correlation spectrum in the physical space is

$$\mathbf{S}_U(\Omega) = \mathbf{Y}\mathbf{s}_q(\Omega)\mathbf{Y}^T$$

The spectral distribution of all quantities linearly related to the degrees of freedom can be computed from the modal cross-correlation. Assume that two quantities $y(t)$ and $z(t)$ are linear functions of the degrees of freedom $\mathbf{u}$, so that

$$y(t) = \mathbf{a}^T\mathbf{u}(t)$$
$$z(t) = \mathbf{b}^T\mathbf{u}(t)$$

Then, by using the definitions above and the linearity of the Fourier transform,

$$S_{yz}(\Omega) = \frac{1}{\pi}\mathrm{F}[R_{xy}(\tau)] = \frac{1}{\pi}\mathrm{F}[\mathbf{a}^T R_{\mathbf{uu}}(\tau)\mathbf{b}] = \mathbf{a}^T\mathbf{S}_U(\Omega)\mathbf{b} = \mathbf{a}^T\mathbf{Y}\mathbf{s}_q(\Omega)\mathbf{Y}^T\mathbf{b}$$

Response Spectrum Analysis Theory

Response spectrum analysis is used for computing an approximation of the structural response to transient, nondeterministic events, such as earthquakes or shocks. The idea is that the event is characterized by the peak response that it would give a single degree of freedom (SDOF) oscillator having a certain natural frequency and damping ratio. This response value is provided as a function of the natural frequency of the oscillator. The actual load history of the event is not known explicitly.

This theory section contains:

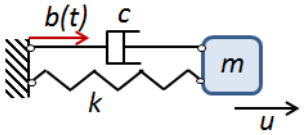
- Single Degree of Freedom System
- Definition of a Response Spectrum
- Solution Using Response Spectrum

- Mode Summation
  - Partitioning into Periodic and Rigid Modes
  - The Gupta Method
  - The Lindley–Yow Method
  - Combining Periodic and Rigid Modes
  - Summing the Periodic Modes
  - SRSS Method
  - Grouping Method
  - Ten Percent Method
  - Double Sum Method
  - Der Kiureghian Correlation Coefficient (CQC)
  - Absolute Value Sum
  - Summing the Rigid Modes
  - Rigid Mode Combination Method A
  - Rigid Mode Combination Method B
  - Missing Mass Correction
  - The Static ZPA Method
- Summation Over Spatial Directions
  - SRSS Method
  - 100–40–40 Method (Percent Method)
  - CQC3 Method
  - SRSS3 Method

SINGLE DEGREE OF FREEDOM SYSTEM

Response spectrum analysis is based on the response of a set of single degree of freedom (SDOF) systems.

Consider a mass–spring–damper system, attached to a moving base. The base movement is $b(t)$.



The equation of motion for the mass can, if there are no external loads, be written as

$$m\ddot{u} + c(\dot{u} - \dot{b}) + k(u - b) = 0$$

Dividing by the mass, and using customary notation,

$$\ddot{u} + 2\zeta\omega_0\dot{u} + \omega_0^2 u = 2\zeta\omega_0\dot{b} + \omega_0^2 b$$

Here, the undamped natural (angular) frequency is

$$\omega_0 = \sqrt{\frac{k}{m}}$$

and the damping ratio is

$$\zeta = \frac{c}{2\sqrt{km}}$$

It can be seen that the support movement acts as a forcing term, and that the solution depends only on two parameters ω_0 and ζ, and not on the individual values of m, c, and k.

Instead of using the absolute displacement as DOF, one can use the relative displacement between the mass and the base,

$$u_r = u - b$$

The equation of motion can then be stated as

$$\ddot{u}_r + 2\zeta\omega_0\dot{u}_r + \omega_0^2 u_r = -\ddot{b} \tag{3-191}$$

In practice, such approach means a frame transformation, where the support movement appears as a gravity-like load.

DEFINITION OF A RESPONSE SPECTRUM

For given values of ω_0, ζ, and $b(t)$, it is a trivial task to solve Equation 3-191 for the whole duration of the event plus some extra time to allow for the response to reach a possible maximum. The acceleration, velocity, and displacement response spectra are defined as

$$S_a(\omega_0, \zeta, b(t)) = \max\left|\ddot{u}(t, \omega_0, \zeta)\right|$$
$$S_v(\omega_0, \zeta, b(t)) = \max\left|\dot{u}(t, \omega_0, \zeta)\right|$$
$$S_d(\omega_0, \zeta, b(t)) = \max\left|u(t, \omega_0, \zeta)\right|$$

These are *absolute* spectra. One can do a similar definition of the *relative* spectra by using instead the relative displacement u_r. It is clear from the definition that there is not a one-to-one relation between the response spectrum and the base acceleration history. The response spectrum gives information about the peak value, but not about when it occurs.

The velocity and acceleration response spectra often are approximated by

$$S_v \approx \omega_0 S_d$$
$$S_a \approx \omega_0^2 S_d$$

Such spectra are called the pseudovelocity spectrum and the pseudoacceleration spectrum, respectively. The expressions contain an assumption about harmonic motion, so that the response is dominated by the homogeneous solution to the equation of motion.

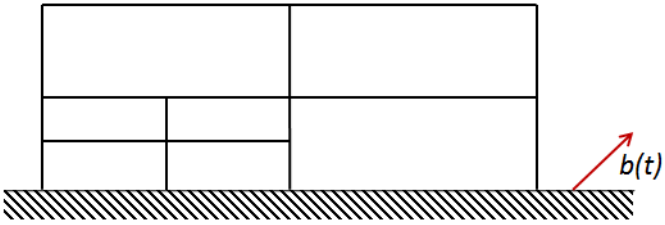
SOLUTION USING RESPONSE SPECTRUM

Assume that one has a structure discretized by FEM, so that the equations of motion on matrix form are

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t) \tag{3-192}$$

Now, let the structure be connected to a common "ground" at a number of points. These points then have a base motion given by

$$\begin{bmatrix} b_x(t) \\ b_y(t) \\ b_z(t) \end{bmatrix}$$



Let $\mathbf{b}(t)$ be a vector that has the same size as the displacement vector $\mathbf{u}$ (the total number of DOF), but it contains only three different values: $b_x(t)$ in all x-translation DOFs, $b_y(t)$ in all y-translation DOF, and $b_z(t)$ in all z-translation DOF. The relative displacement is now $\mathbf{u}_r = \mathbf{u} - \mathbf{b}$. With no external load, Equation 3-192 can be written as

$$\mathbf{M}(\ddot{\mathbf{u}}_r + \ddot{\mathbf{b}}) + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = \mathbf{0}$$

or

$$\mathbf{M}\ddot{\mathbf{u}}_r + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = -\mathbf{M}\ddot{\mathbf{b}}$$

Then, the fact that a rigid body motion does not give any elastic or viscous forces has been used, so that

$$\mathbf{K}\mathbf{b} = \mathbf{0}$$
$$\mathbf{C}\dot{\mathbf{b}} = \mathbf{0}$$

By solving the undamped eigenvalue problem

$$(\mathbf{K} - \omega^2\mathbf{M})\varphi = \mathbf{0}$$

with the grounded nodes being fixed, a set of eigenmodes that can represent the relative displacements is obtained. Since the mode displacements are zero at the support points, it is however clear that the modes cannot represent the absolute displacements, for which the support points are moving.

By standard operations for mode superposition (and assuming mass matrix normalization and a diagonalizable damping matrix) the decoupled modal equations are

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2 q_j = -\phi_j^T \mathbf{M}\ddot{\mathbf{b}} \tag{3-193}$$

The equation is similar to the standard SDOF system in Equation 3-191.

The assumption about mass matrix normalization is not essential, but some expressions are simplified. In COMSOL Multiphysics, the computed modal participation factors are consistent with the chosen mass matrix normalization. See also Modal Participation Factors.

In Equation 3-193, q_j is the modal coordinate for mode j, so that the relative displacement can be written as a superposition of the eigenmodes φ_j.

$$\mathbf{u}_\mathrm{r} = \sum_j \varphi_j q_j = \Phi\mathbf{q}$$

The support motion can be written as

$$\mathbf{b}(t) = b_x(t)\mathbf{1}_x + b_y(t)\mathbf{1}_y + b_z(t)\mathbf{1}_z \tag{3-194}$$

The notation $\mathbf{1}_x$ mean a vector that has the value 1 in all DOF representing x-translation, and the value 0 in all other DOF. Inserting Equation 3-194 in Equation 3-193 gives

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2 q_j = -\Gamma_{xj}\ddot{b}_x(t) - \Gamma_{yj}\ddot{b}_y(t) - \Gamma_{zj}\ddot{b}_z(t)$$

The multipliers Γ_{kj} are the modal participation factors defined as

$$\Gamma_{kj} = \phi_j^T \mathbf{M}\mathbf{1}_k$$

Thus, the maximum amplitude of mode j, when loaded by a base motion described by a response spectrum in direction k, is

$$\hat{q}_{kj} = S_\mathrm{d}(\omega_j, \zeta_j, b_k(t))\Gamma_{kj} = S_{\mathrm{d},k}\Gamma_{kj}$$

or, using the pseudoacceleration spectrum

$$\hat{q}_{kj} = \frac{S_\mathrm{a}(\omega_j, \zeta_j, b_k(t))\Gamma_{kj}}{\omega_j^2} = \frac{S_{\mathrm{a},k}\Gamma_{kj}}{\omega_j^2}$$

In practice, several modes will have natural frequencies in the frequency range covered by the Design Response Spectrum. This means that a superposition is needed. There

are several rules for how this superposition can be done, as will be described in detail below.

MODE SUMMATION

The summation rules are nonlinear. Thus, all result quantities must be summed based on its own modal response. For example, stress components are computed using the modal stresses and cannot be recovered from the response spectrum superposition of displacements.

The general approach is to consider the excitation in three directions, I, (I = 1,2,3) separately. First all modal responses are summed for each direction, and then the results for the three directions are summed. Some methods, however, do both combinations in one sweep.

In a high frequency mode, the mass of the SDOF oscillator will mainly be translated in phase with the support. Such modes constitute the *rigid modes*. Their responses are synchronous with each other (and with the base motion). This means that for rigid modes, a pure summation should be used, since they are fully correlated.

Modes with a significant dynamic response constitute the periodic modes. The maximum value for such modes will be more or less randomly distributed in time since their periods differ. For this reason, the periodic part of the response requires more sophisticated summation techniques. A plain summation of the maximum values will in general significantly overestimate the true response.

Modes which are in a transition region will partially contribute to the periodic modes, and partially to the rigid ones.

In addition, it is sometimes necessary to add some static load cases, containing a *missing mass correction*. The reason is that when only a limited set of eigenmodes is used in the superposition, those modes do not represent the total mass of the structure.

In the following, R_I denotes any result quantity caused by excitation in direction I. R_I can be for example be displacement, velocity, acceleration, strain component, stress component, equivalent stress, or a beam section force. The periodic part of R_I is denoted $R_{\mathrm{p}I}$, and the rigid part is denoted $R_{\mathrm{r}I}$. Similarly, $R_{\mathrm{p}I,j}$ and $R_{\mathrm{r}I,j}$ denote the results from an individual eigenmode j.

Not all analyses require a separation into periodic and rigid modes. In such case, all modes are treated as periodic.

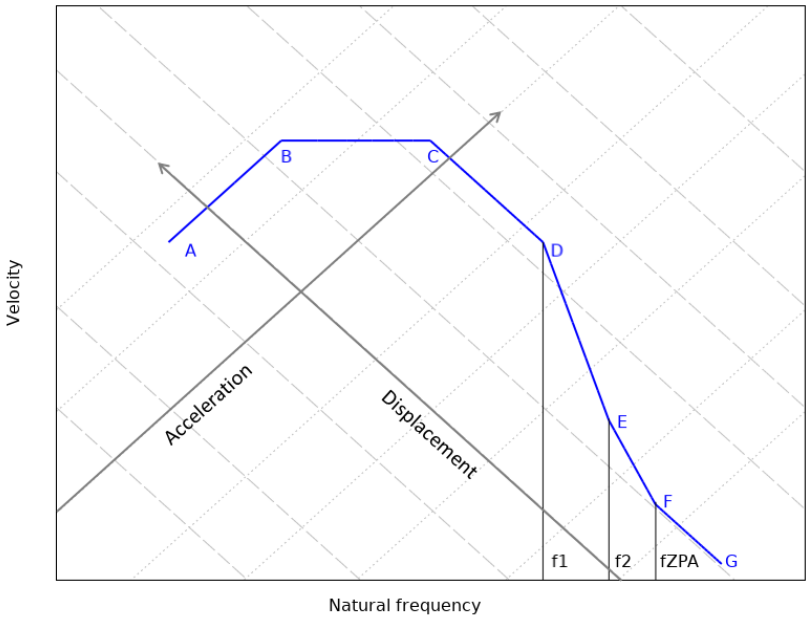


Figure 3-38: A schematic tripartite plot of a design response spectrum. Both axes have logarithmic scales.

There are two different methods in use, by which the partitioning can be done. In either case, for mode j,

$$R_{\mathrm{r}I,\mathrm{j}} = \alpha_j R_{I,j}$$
$$R_{\mathrm{p}I,\mathrm{j}} = \sqrt{1-\alpha_j^2}\, R_{I,j}$$

This definition has the property that

$$R_{I,\mathrm{j}}^2 = R_{\mathrm{p}I,j}^2 + R_{\mathrm{r}I,j}^2$$

The difference between the two methods lies in how the coefficients α_j are determined. For low frequencies, it should approach the value 0 (fully periodic modes). And for high frequencies, the value is 1 (fully correlated rigid modes).

The Gupta Method

In the Gupta method, α_j is a linear function of the logarithm of the natural frequency.

$$\alpha_j = 0 \qquad f_j \le f_1$$
$$\alpha_j = \frac{\log(f_j/f_1)}{\log(f_2/f_1)} \qquad f_1 < f_j < f_2$$
$$\alpha_j = 1 \qquad f_j \ge f_2$$

where f_1 and f_2 are two key frequencies. Thus, for eigenfrequencies below f_1, the modes are considered as purely periodic, and above f_2 as purely rigid. In the original Gupta method, the lower key frequency is given by

$$f_1 = \frac{S_{\mathrm{a,max}}}{2\pi S_{\mathrm{v,max}}}$$

where $S_{\mathrm{a,max}}$ and $S_{\mathrm{v,max}}$ are the maximum values of the acceleration and velocity spectra, respectively. In the idealized spectrum shown in Figure 3-38, this exactly matches the point D.

The second key frequency should be chosen so that the modes above this frequency behave as rigid modes. The frequency can be taken as the one where response spectra for different damping ratios converge to each other.

The Lindley–Yow Method

In the Lindley–Yow method, the coefficient α_j depends directly on the response spectrum values, and not only on the frequency. As a consequence, it is possible that a certain mode can be considered as having a different degree of rigidness for different excitation directions

$$R_{\mathrm{p}I,\mathrm{j}} = \frac{\sqrt{S^2_{\mathrm{a},I}(f_j) - S^2_{\mathrm{ZPA},I}}}{S_{\mathrm{a},I}(f_j)} R_{I,\mathrm{j}}$$

The so-called *zero period acceleration* (ZPA) is the maximum ground acceleration during the event

$$S_{\mathrm{ZPA},I} = \max_t \left| \ddot{b}_I(t) \right|$$

This is the high frequency asymptotic value of the absolute acceleration (or pseudoacceleration). It also corresponds to the F-G part of the spectrum in Figure 3-38.

Thus,

$$\alpha_{j,I} = \frac{S_{\mathrm{ZPA},I}}{S_{\mathrm{a},I}(f_j)}$$

The value of α_j must be in the range from 0 to 1, and it must increase with frequency. For this reason, NRC RG 1.92 requires that α_j must be set to zero for any eigenmodes below point C in Figure 3-38. For a general spectrum, this is implemented as a strict requirement that α_j has a monotonous decrease with decreasing frequency from f_{ZPA}. As soon as an increase in α_j is found, the value is set to zero for all lower frequencies.

Combining Periodic and Rigid Modes

Once the periodic and rigid responses have been summed up separately, they are combined as

$$R_I = R_{\mathrm{p}I}^2 + R_{\mathrm{r}I}^2$$

Summing the Periodic Modes

All summation rules for periodic modes except Absolute Value Sum and NRL Method can be summarized by the following expression:

$$R_{\mathrm{p}I}^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} R_{\mathrm{p}I,i} R_{\mathrm{p}I,j}$$

Here, $R_{\mathrm{p}I}$ is the total periodic response of some result quantity R with respect to excitation in direction I ($I = 1, 2, 3$). $R_{\mathrm{p}I,j}$ is the result from an individual eigenmode j, and N modes are used in the summation. The interaction between the modes is determined by the coefficient C_{ij} ($0 \le C_{ij} \le 1$). The different evaluation methods vary only in the definition of C_{ij}.

Since C_{ij} is symmetric and $C_{ij} = 1$ when $i = j$, it is more efficient to use the expression

$$R_{\mathrm{p}I}^2 = \sum_{i=1}^{N} R_{\mathrm{p}I,i}^2 + 2\sum_{i=1}^{N}\sum_{j=i+1}^{N} C_{ij} R_{\mathrm{p}I,i} R_{\mathrm{p}I,j}$$

The result quantity R is computed using the ordinary definitions of how a variable is obtained from the DOF fields. This can be expressed as $R = g(\mathbf{u})$. The operator g is however applied to the mode shape, multiplied by a scalar (spectrum value times participation factor)

$$R_{I,j} = g(S_{\mathrm{d},I}\Gamma_{Ij}\phi_j)$$

Next, the different evaluation methods are presented.

SRSS Method

In the SRSS method, it is assumed that the modes are statistically independent, so that

$$C_{ij} = 1 \quad i = j$$
$$C_{ij} = 0 \quad i \neq j$$

Grouping Method

The modes are grouped according to the following rule:

1 Start a new group m, by inserting the lowest, not yet grouped eigenmode k.

2 Step up through the eigenfrequencies from k.

3 As long as $f_i \leq 1.1 f_k$, add mode i to the group.

4 When **3** is not fulfilled any longer, go back to **1**.

There are now a number of groups (where some could contain just a single eigenmode), and the coupling coefficient are defined as

$$C_{ij} = \mathrm{sgn}(R_{\mathrm{p}I,i} R_{\mathrm{p}I,\mathrm{j}}) \quad \text{when the modes are in the same group}$$
$$C_{ij} = 0 \quad \text{when the modes are not in the same group}$$

The use of the sign function indicates that the term is always added with a positive sign, that is a cross term can never decrease the total sum. There is also an option to allow signed contributions, so that

$$C_{ij} = 1 \quad \text{when the modes are in the same group}$$
$$C_{ij} = 0 \quad \text{when the modes are not in the same group}$$

Ten Percent Method

The ten percent method is similar to the grouping method, and has the following definition:

$$C_{ij} = \mathrm{sgn}(R_{\mathrm{p}I,i} R_{\mathrm{p}I,\mathrm{j}}) \quad f_i \leq 1.1 f_j$$
$$C_{ij} = 0 \quad \text{else}$$

As can be seen, the ten percent method will always give a higher value than the grouping method, since all pairs that are inside a group will also fulfill the criterion for

including the cross term. As in the previous method, there is also an option to use summation with signs, in which case

$$C_{ij} = 1 \quad f_i \le 1.1 f_j$$
$$C_{ij} = 0 \quad \text{else}$$

Double Sum Method

In the double sum method, the correlation between two modes depends on three factors:

- The frequency spacing between the two modes
- The damping
- The duration of the event

Note that there exist two distinct versions of this method. In NRC Regulatory Guide 1.92 revision 1, the mode correlation coefficient is given by

$$C_{ij} = \operatorname{sgn}(R_{\mathrm{p}I,i} R_{\mathrm{p}I,\mathrm{j}}) \left[1 + \left(\frac{f_i' - f_j'}{\zeta_i' f_i + \zeta_j' f_j} \right)^2 \right]^{-1}$$

whereas in revision 2 and later version of the same Regulatory Guide the expression is

$$C_{ij} = \left[1 + \left(\frac{f_i' - f_j'}{\zeta_i' f_i + \zeta_j' f_j} \right)^2 \right]^{-1}$$

The latter expression should be considered as more correct and in line with the original theory.

The modified frequency, f_i', is defined as

$$f_i' = f_i \sqrt{1 - \zeta_i^2}$$

where ζ_i is the modal damping. In the implementation in COMSOL Multiphysics, all modes are assumed to have the same damping.

The modified damping, ζ_i, is defined as

$$\zeta_i' = \zeta_i + \frac{1}{\pi t_{\mathrm{d}} f_i}$$

where t_{d} is a separate input, called the time of duration. The value of ζ_i differs between modes, even though the damping is constant.

Der Kiureghian Correlation Coefficient (CQC)

This method, which is often called *complete quadratic combination* (CQC), is similar to the previous double sum method. The general expression contains modal damping values. Given that a single damping value is used here, the mode correlation expression can be simplified to

$$C_{ij} = \frac{8\zeta^2\sqrt{f_i f_j}(f_i + f_j)f_i f_j}{(f_i^2 - f_j^2)^2 + 4\zeta^2 f_i f_j (f_i + f_j)^2} \tag{3-195}$$

Absolute Value Sum

This is the most conservative method, where the peak responses for all modes are summed

$$R_{pI} = \sum_{i=1}^{N} |R_{pI,i}|$$

This would happen only if all modes reached their peaks simultaneously.

NRL Method

The NRL rule differs from the other summation rules in that the mode with the highest result is treated separately. The remaining modes are handled using an SRSS type expression. Let

$$R_{pI,\max} = \max_i(|R_{pI,i}|) = |R_{pI,m}|$$

where m is the index for the mode in which the maximum occurs. Then the NRL summation stipulates that

$$R_{pI} = R_{pI,m} + \sqrt{\sum_{i=1}^{N} R_{pI,i}^2 - R_{pI,m}^2}$$

In the NRL method, it is possible to take the effect of closely spaced modes into account. Modes that are close in frequency tend to reach their peaks close in time, and may then, to some extent cancel each other. The Closely Spaced Method (CSM) combines two or more closely spaced modes into a single effective mode. In order to be closely spaced, the modes shall have frequencies that are within 10% from their common mean frequency.

$$f_j > \frac{f_j + f_k}{2}$$
$$f_k < \frac{f_j + f_k}{2}$$

Let

$$B_{I,i} = e^{\frac{\zeta\pi}{2}} R_{\mathrm{p}I,i}$$

be a damping corrected modal amplitude. Here, ζ is the relative damping, usually set to 0.02. Note that the multiplying factor is slightly larger than 1; (1.032 when $\zeta = 0.02$). In principle, this correction should be applied to the modal amplitude, ut for any quantity that has a linear dependency on the mode shape, the multiplier can be directly applied to that quantity.

The envelope of the combined response for two modes (j and k) is

$$E(t) = e^{-at}\sqrt{(B_{I,j} + B_{I,k})^2 - 4B_{I,j}B_{I,k}\sin(dt)^2}$$

where

$$a = \zeta\frac{\Omega_j + \Omega_k}{2} = \zeta\pi(f_j + f_k)$$

and

$$d = \frac{\sqrt{1-\zeta^2}\left|\Omega_j - \Omega_k\right|}{2}$$

It can then be shown that the times when the response combine in a maximal way is given by a set of time t_{n}. The first possible time is $t_0 = 0$. The following are given by

$$t_{\mathrm{n}} = \frac{\operatorname{asin}\left[\dfrac{-a(B_{I,j}^2 + B_{I,k}^2)}{2(B_{I,j}B_{I,k})\sqrt{a^2 + d^2}}\right] - \theta}{2d}$$

where

$$\theta = \operatorname{atan}\left(\frac{a}{d}\right) \quad \text{with} \quad 0 \le \theta \le \pi/2$$

Since the inverse sine function has multiple solutions, there can be several possible times when the maximum occurs. Due to the damping, only the first ones can be a true maximum, though.

Let S be the argument to the inverse sine function above.Then there are some different cases:

S>1

The maximum occurs at t=0.

$E_{\mathrm{I}} = E(0)$

S=1

One more possible solution exists, at t_1.

$E_{\mathrm{I}} = \max(E(0), E(t_1))$

0<S<1

Three time instances must be checked

$E_{\mathrm{I}} = \max(E(0), E(t_1))E(t_2)$

S<0

The modes will not cancel each other, so the method cannot be used.

Finally, the modified NRL sum is written as

$$R_{\mathrm{p}I} = \max(R_{\mathrm{p}I,m}, E_I) + \sqrt{\sum_{i=1}^{N} R_{\mathrm{p}I,i}^2 + E_I^2 - R_{\mathrm{p}I,j}{}^2 - R_{\mathrm{p}I,\mathrm{k}}{}^2 - (\max(R_{\mathrm{p}I,m}, E_I))^2}$$

When computing the maximum value $R_{\mathrm{p}I,m}$, the modes j and k should be excluded. That is, $m \neq j$ and $m \neq k$. The interpretation of this expression is that modes j and k are combined into a new 'effective' mode, which is then treated as any other mode while the original modes j and k are removed.

Summing the Rigid Modes

There are two possible combination methods for summing the rigid modes. The method is chosen implicitly, depending on other settings.

Rigid Mode Combination Method A

This is the more common method. The rigid modes are summed algebraically as

$$R_{\mathrm{r}I} = \sum_{i=1}^{N} R_{\mathrm{r}I,i} + R_{\mathrm{mm},I}$$

where $R_{\mathrm{mm},I}$ is the term for the missing mass correction, if used.

Rigid Mode Combination Method B

This method is only used when the Lindley–Yow method is used together with the Static ZPA missing mass correction. In this case, the whole rigid mode contribution comes from the static load case, so that

$$R_{\mathrm{r}I} = R_{\mathrm{StaticZPA},I}$$

Missing Mass Correction

In general, a mode superposition using a limited number of modes will miss some mass. With the assumption that the higher-order modes do not have any dynamic amplification, it is possible to device a correction by solving some extra static load cases, containing the acceleration excitation acting on the "lost" mass. So-called *static correction* can be used for mode superposition in general. For the case of response spectrum analysis, the expressions are somewhat simplified.

In terms of the assembled finite element equations, the static correction load $\mathbf{f}_{\mathrm{c}}$ can be written as

$$\mathbf{f}_{\mathrm{c}} = \mathbf{f} - \sum_{k} r_k(\mathbf{M}\phi_k)$$

Here, $\mathbf{f}$ is the original load vector, $\mathbf{M}$ is the mass matrix, and r_k are the modal loads, given by the projection of the load vector on the eigenmodes ϕ,

$$r_k = \phi_k^T \mathbf{f}$$

In the base excitation context when response spectrum analysis is used,

$$\mathbf{f}_I = -\mathbf{M}\ddot{b}_I(t)\mathbf{1}_I$$

so the modal load is

$$r_{k,I} = -\Gamma_{Ik}\ddot{b}_I(t)$$

Thus, the missing mass load is

$$\mathbf{f}_{\mathrm{c},I} = -\mathbf{M}\ddot{b}_I\mathbf{1}_I + \sum_k \Gamma_{Ik}\ddot{b}_I(\mathbf{M}\phi_k)$$

For the rigid body modes, the maximum ground acceleration during the event is equal to the ZPA. The static load is thus

$$\mathbf{f}_{\mathrm{c},I} = S_{\mathrm{ZPA}}\mathbf{M}\left(-\mathbf{1}_I + \sum_k \Gamma_{Ik}\phi_k\right) \tag{3-196}$$

The extra displacement correcting for the missing mass is then given by the standard stationary problem

$$\mathbf{K}\mathbf{u}_{\mathrm{rc},I} = \mathbf{f}_{\mathrm{c},I}$$

To actually compute the load in Equation 3-196, the participation factors from a corresponding eigenfrequency study step are needed. The structure of the load is similar to that of a gravity load, but with the acceleration of gravity replaced by the space-dependent field

$$\sum_k \Gamma_{Ik}\phi_k - \mathbf{1}_I \tag{3-197}$$

The load is implemented by using **Gravity** nodes in the Structural Mechanics interface. The sum of the products between participation factors and mode shapes is performed in a **Combined Solutions** study step.

The Static ZPA Method

In this method, there is no need to deduce the actual missing mass. It can only be used together with the Lindley–Yow method. According to the Lindley–Yow method, all rigid modes have acceleration S_{ZPA}. This acceleration is given to the whole structure. The static load cases are thus just pure gravity loads, but scaled by S_{ZPA} instead of the acceleration of gravity.

SUMMATION OVER SPATIAL DIRECTIONS

The three orthogonal directions in which the design response spectrum is applied cannot, in general, be chosen arbitrarily. The structure may be more susceptible to excitation in a certain direction.

For earthquakes, it is usually assumed that the excitations in the three orthogonal directions are statistically independent. In most cases, there is no reason to assume that the excitations in the two horizontal directions have different spectral properties. Thus, a single design response spectrum is used in the two horizontal directions, and a different one is used in the third vertical (Z) direction.

Often, it is reasonable to assume that the excitations in the two horizontal directions have different amplitudes, even though they share the same spectral properties. The spectrum in the local Y direction is then a scaled version of the spectrum in the local X direction.

$$S_Y = \gamma S_X \quad 0 < \gamma \leq 1$$

The X direction is not a property of the geographical location, but should be chosen as the one giving the worst case for a certain structure. The loading direction which causes the highest response may however not be the same for different result quantities, or for different locations in the structure. For some structures, there is an obvious "weak" direction which can then be chosen as X direction. More often, this is not the case. There are then three possible approaches:

- Use the same spectrum in both horizontal directions — that is, $\gamma = 1$. This will be a conservative approach.
- Run a number of separate analyses where the X direction is rotated to different orientations. If 15 degrees can be considered as a small enough rotation increment, then seven analyses are needed.
- Use a combination rule (CQC3), which takes the possible rotation into account.

SRSS Method

In the SRSS (square root of sum of squares) method, the total resultant is computed as

$$R = \sqrt{\sum_{I=1}^{3} R_I^2}$$

This expression contains an assumption of a statistical independence between the peak responses in three directions.

100–40–40 Method (Percent Method)

In this method, the contribution from the worst direction is taken at full value, whereas the two other contributions are reduced. There are two variants in commonly in use, the 40% (100–40–40) method and the 30% (100–30–30) method. The interpretation

is clear: at the time when the peak values is reached in the worst direction, the values in the other direction are not higher than 40% (or 30%) of their individual peak values.

Let the response for the three directions be reordered so that

$$|R_1| \geq |R_2| \geq |R_3|$$

The total response for the 40% method is then computed as

$$R = |R_1| + 0.4|R_2| + 0.4|R_3|$$

In some formulations of this rule, the renumbering is not done, and the expression is written instead as

$$R = \begin{cases} \pm[R_X \pm 0.4R_Y \pm 0.4R_Z], \text{ or} \\ \pm[R_Y \pm 0.4R_Z \pm 0.4R_X], \text{ or} \\ \pm[R_Z \pm 0.4R_X \pm 0.4R_Y] \end{cases}$$

In practice, the same result is obtained as long as signs are properly taken into account when summing the results for multiple responses.

The 40% method is mostly slightly conservative when compared to the SRSS summation. The 30% method is significantly less conservative, and it will often give lower predictions than the SRSS method.

The percent methods are not spatially isotropic. For a symmetric structure, members which for symmetry reason should have the same level of loading will not experience that. The orientation of the reference axes for the acceleration orientation will matter.

CQC3 Method

The CQC3 method extends the CQC principles also to the spatial combination. In the CQC3 method, the modal and spatial combination are performed simultaneously. It is however only formally applicable if only the periodic modes are taken into account.

As in the standard CQC method, the modal response for each loading direction is summed as

$$R_{\mathrm{p}I}^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} R_{\mathrm{p}I,i} R_{\mathrm{p}I,j}$$

where the Der Kiureghian expression Equation 3-195 for C_{ij} is used.

In addition, a similar expression giving the cross coupling between the responses to the spectra in the two horizontal directions is formed:

$$R_{pXY}^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} R_{pX,i} R_{pY,j}$$

It is now conceptually assumed that the response spectra are instead applied in a local coordinate system X'-Y', which is rotated an angle θ with respect to the X-Y orientations. It can then be shown that

$$R_{pX',i} = R_{pX,i}\cos\theta + R_{pY,i}\sin\theta$$
$$R_{pY',i} = R_{pY,i}\cos\theta - R_{pX,i}\sin\theta$$

Also, if the relation between the two spectra in the horizontal plane is such that

$$S_Y = \gamma S_X$$

the same ratio γ will apply to the responses. The peak response as function of the rotation angle is obtained by an SRSS type summation

$$R(\theta) = \{[R_{pX}^2 + \gamma^2 R_{pY}^2]\cos^2\theta + [R_{pY}^2 + \gamma^2 R_{pX}^2]\sin^2\theta + 2\sin\theta\cos\theta(1-\gamma^2)R_{pXY}^2 + R_{pZ}^2\}^{1/2}$$

It can be seen that for $\gamma = 1$, the standard SRSS expression is retrieved.

The angle θ_{max} giving the maximum response $R(\theta_{max})$ turns out to be independent of γ, and it has the value

$$\theta_{max} = \frac{1}{2}\operatorname{atan}\left[\frac{2R_{pXY}^2}{R_{pX}^2 - R_{pY}^2}\right]$$

There are two roots for θ_{max}, both of which must be checked.

The attractiveness of the CQC3 method is that the same spectrum can be applied to an arbitrary pair of orthogonal axes. The scaling of the secondary spectrum, as well as the orientation of the worst direction, is taken care of by the method.

Extending to Rigid Modes

As mentioned above, the original CQC3 method only deals with the periodic part of the solution, so it is limited to cases dominated by such modes. It is however possible to make an extension taking also the rigid (high frequency) modes.

Studying how the rigid modes enter the problem when using CQC (or any of the similar combination rules) together with SRSS spatial combination gives some insight:

$$R^2 = \sum_{I=1}^{3} R_I^2 = \sum_{I=1}^{3} (R_{\mathrm{p}I}^2 + R_{\mathrm{r}I}^2) = \sum_{I=1}^{3} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} R_{\mathrm{p}I,i} R_{\mathrm{p}I,j} + R_{\mathrm{r}I}^2 \right)$$

Thus, the rigid responses enter the final results as an extra mode, not coupled to the periodic modes. Define a cross term also for the rigid response

$$R_{\mathrm{r}XY}^2 = R_{\mathrm{r}X} R_{\mathrm{r}Y}$$

Now, it is possible to have also the rigid modes in a CQC3 context. Another way of expressing this is that the rigid response is treated as mode $N + 1$, the summation in the CQC3 rule is extended to $N + 1$, and

$$\begin{cases} C_{N+1\,N+1} = 1 \\ C_{i\,N+1} = 0 \quad \text{when} \quad i \neq N+1 \end{cases}$$

In the GUI, this extension is selected by checking the **Augment with rigid response** checkbox.

SRSS3 Method

The SRSS3 method is a special case of the CQC3 rule, in which the mode correlation is ignored, that is

$$\begin{aligned} C_{ij} &= 1 \quad i = j \\ C_{ij} &= 0 \quad i \neq j \end{aligned}$$

It retains the property of selecting the worst orientation, through the search for $\theta_{\max}$. The extension to rigid modes is the same as for CQC3.

ADDING A STATIC SOLUTION

For the final evaluation, it is common that a static solution needs to be added to the response spectrum results. A common case is the inclusion of the effects of self-weight.

Since the response spectrum solution does not contain any information about directions, while the static solution in general contains both positive and negative values for any given quantity, some care is needed when performing the summation. In most cases, a reasonable approach is to look at the sign of the static solution, and then add the response spectrum solution using the same sign. Let R be any result quantity from a response spectrum analysis, computed as described above, and R_s be the same result quantity in the static solution. Then the total result R_t is

$$R_t = R_s + R\,\mathrm{sgn}(R_s)$$

This is the default option, which you get when **Sign of response spectrum solution** is set to **Same as static solution**. There are two other options, **Positive** and **Negative**, in which cases the total result is computed as

$$R_t = R_s + R$$

and

$$R_t = R_s - R$$

respectively.

Note that what is here called the static solution actually can be any type of real-valued solution, for example, the solution at a certain time step of a time-dependent study.

For information about how to perform a response spectrum analysis, see Response Spectrum Analysis Study

Linear Buckling

The most common linear buckling analysis consists of two steps. First, a stationary problem is solved using a unit load of arbitrary size. The critical load is then obtained by solving an eigenvalue problem, where the eigenvalue λ is the multiplier to the original load that would cause buckling.

- For a general introduction, see Linearized Buckling Analysis.
- For the theory of buckling analysis of periodic structures, see Cyclic Symmetry and Floquet Periodic Conditions.

The formulation in terms of virtual work is

$$\delta W = \int_V (-\delta\varepsilon : \sigma - \lambda\delta(\varepsilon_{GL} - \varepsilon) : \sigma) dV = 0 \quad (3\text{-}198)$$

Here, ε is the engineering strain, ε_{GL} is the Green–Lagrange strain. In terms of stiffness matrices, this corresponds to

$$(K_L + \lambda K_{NL}(\mathbf{u}_0))\mathbf{u} = 0$$

where K_L is the linear stiffness matrix, and K_{NL} is the nonlinear contribution to the full stiffness matrix. The symbolic linearization point $\mathbf{u}_0$ is the displacement vector caused by the unit load.

Strictly speaking, this formulation assumes that geometric nonlinearity is not used in the eigenvalue step. The Green–Lagrange tensor is inserted explicitly in the second term of Equation 3-198, while the first term uses the linear (engineering) strain tensor.

If, however, geometric nonlinearity is selected in the linear buckling study step, Equation 3-198 is replaced by

$$\delta W = \int_V (-\delta\varepsilon_{GL} : \sigma - (\lambda - 1)\delta(\varepsilon_{GL} - \varepsilon) : \sigma) dV = 0$$

By using the term $(\lambda - 1)$, the effect of using the Green–Lagrange strain tensor in the first term is to a large extent removed. Unless the unit load is significantly larger than the buckling load, the result will be the same as the intended, even if geometric nonlinearity was inadvertently selected in the eigenvalue study step.

Damping

For dynamic problems, the damping of the structure is usually an important property.

In this section:

- Rayleigh Damping
- Loss Factor Damping
- Viscous Damping
- Maximum Loss Factor
- Wave Attenuation

Rayleigh Damping

Rayleigh damping is described by two coefficients: the mass damping coefficient α_{dM} and the stiffness damping coefficient β_{dK}. Rayleigh damping will give the following contribution to the virtual work

$$\delta W = \int_V \left(-\delta(\nabla \mathbf{u}) : \beta_{dK} \frac{\partial P}{\partial t} - \alpha_{dM} \rho \delta \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} \right) dv$$

Here P is the first Piola–Kirchhoff stress tensor.

Since Rayleigh damping is added directly to the virtual work equation, it does not affect the constitutive relation. As a consequence, the stresses and strains will for a linear elastic material still be in phase. This stands in contrast to the other damping models.

Rayleigh damping can be used both in the time domain and in the frequency domain.

Rayleigh damping is not directly related to any physical property. Historically, it was introduced since it was numerically attractive to have a damping matrix which was a linear combination of the mass and stiffness matrices

$$C = \alpha M + \beta K$$

This operation is usually implied to be done at the global assembled matrix level. Such an interpretation is however only meaningful for pure structural mechanics problems, but not in a general multiphysics context. For this reason, Rayleigh damping in COMSOL Multiphysics is a material property, rather than a global property of the system of equations. Enter the same Rayleigh damping parameters for all materials, and solve a pure structural mechanics problem to retrieve the classical definition.

Loss Factor Damping

Loss factor damping is only applicable in frequency domain. When using loss factor damping, a complex constitutive matrix is used. With an isotropic loss factor η_s, this means that

$$D^c = (1 + j\eta_s)D$$

where D is the constitutive matrix computed from the material data, and D^c is the complex constitutive matrix used when computing the stresses. For a linear elastic material, this would be equivalent to multiplying Young's modulus by the factor $(1 + j\eta_s)$. For a nonlinear elastic material, this applies to the tangential stiffness.

It is also possible to give individual loss factors for each entry in the constitutive matrix, so that

$$D^c_{mn} = (1 + j\eta_{s,\,mn})D_{mn}$$

In the case of an orthotropic material, yet another option is available, where each individual component of Young's modulus and shear modulus can be given an individual loss coefficient:

$$E_m^c = (1 + j\eta_{E,m})E_m$$
$$G_m^c = (1 + j\eta_{G,m})G_m$$
$$m = 1, 2, 3$$

The complex moduli are then used to form the constitutive matrix.

For hyperelastic materials, the loss information appears as a contribution to the second Piola–Kirchhoff stress:

$$S_q = j\eta_s \frac{\partial W_s}{\partial E}$$

For loss factor damping, the following definition is used for the elastic part of the entropy:

$$S_{elast} = \alpha : (s - j\eta_s(\mathbf{C} : \varepsilon))$$

note that S_{elast} here denotes the entropy contribution and not any stress.

This is because the entropy is a function of state and thus independent of the strain rate, while the damping represents the rate-dependent effects in the material (for example, viscous or viscoelastic effects). The internal work of such inelastic forces averaged over the time period $2\pi/\omega$ can be computed as

$$Q_h = \frac{1}{2}\omega\eta_s \text{real}(\varepsilon : \text{conj}(\mathbf{C} : \varepsilon))$$

Q_h can be used as a heat source for modeling of the heat generation in vibrating structures, when coupled with the frequency-domain analysis for the stresses and strains.

Viscous Damping

Viscous damping can be added to the material models. It will cause an extra stress proportional to the rate of elastic strain in the material,

$$s_q = \eta_b \dot{\varepsilon}_{el,vol} + 2\eta_v \dot{\varepsilon}_{el,dev}$$

where η_b and η_v are the bulk and shear viscosity coefficients, respectively.

Viscous damping can be used in both frequency and time domain analyses.

In case of geometric nonlinearity, the viscous stress is treated as being a Cauchy stress acting in the actual configuration (spatial frame). The resulting contribution to the second Piola–Kirchhoff stress is calculated as

$$S_q = \left(\eta_b - \frac{2}{3}\eta_v\right)\dot{J}_{el}C_{el}^{-1} + 2\eta_v J_{el}C_{el}^{-T}\dot{E}_{el}C_{el}^{-1}$$

where C_{el} is the elastic Cauchy–Green tensor, $\dot{E}_{el}$ is the rate of the elastic Green–Lagrange strain, and J_{el} is the elastic volume ratio.

When using artificial viscosity for shock capturing (Ref. 52), the bulk and shear viscosities are nonlinear functions of the strain rate

$$\eta_b = \rho L_e(c_L c + c_Q|\Delta\dot{u}|)$$

$$\eta_v = \frac{3}{4}\rho L_e(c_L c + c_Q|\Delta\dot{u}|)$$

where c_L and c_Q are linear and quadratic tuning parameters, L_e is the typical length of the element, ρ is the density, and c is the wave speed. The variable

$$\Delta\dot{u} = L_e\dot{\varepsilon}_{vol}$$

is a measure of the shock amplitude, and it can be restricted to compressive strains only.

Maximum Loss Factor

Maximum Loss Factor is a special damping model which can be used in frequency as well as in time domain studies. The model approximates a target loss factor, η_{max}, when the material is loaded at a given reference frequency, f_{ref}. The formulation of this damping model is very similar to the *Standard Linear Solid* (SLS) viscoelastic model with both deviatoric and viscoelastic strains. The SLS model is often represented with a spring–dashpot analogy where an additional branch is coupled in parallel to the parent material model (for instance a *Linear Elastic Material*). The loss factor is defined as

$$\eta = \frac{G''}{G'} = \frac{K''}{K'}$$

where G'' and K'' are the shear and bulk loss moduli, and G' and K' are the shear and bulk storage moduli, respectively. The parallel coupled SLS branch adds an inelastic stress, S_q.

$$S_q = K_v \mathrm{tr}(\varepsilon_{el} - \varepsilon_v)\mathbf{I} + 2G_v \mathrm{dev}(\varepsilon_{el} - \varepsilon_v)$$

Here, the auxiliary viscous strain, ε_v, in the SLS branch follows the relation

$$\tau_v \dot{\varepsilon}_v + \varepsilon_v = \varepsilon_{el}$$

The shear and bulk viscoelastic moduli and the relaxation time are derived from the target maximum loss factor, η_{max}, and the reference frequency, f_{ref}

$$G_v = \frac{2\eta_{max}}{\omega_{ref}\tau_v} G \quad \text{and} \quad K_v = \frac{2\eta_{max}}{\omega_{ref}\tau_v} K$$

$$\tau_v = \frac{\sqrt{\eta^2_{max} + 1} + \eta_{max}}{\omega_{ref}} \quad \text{where} \quad \omega_{ref} = 2\pi f_{ref}$$

Here G and K are the equivalent shear and bulk moduli of the parent material model, for instance a *Linear Elastic Material*. The loss factor is then defined from the frequency-dependent expression

$$\eta(\omega) = 2\eta_{max} \frac{\omega/\omega_{ref}}{1 + (\omega/\omega_{ref})^2}$$

The loss factor equals the target maximum loss factor when the material is loaded with a harmonic load at the reference frequency

$$\eta(\omega_{ref}) = \eta_{max}$$

Wave Attenuation

For linear elastic isotropic material, the effective bulk and shear viscosity coefficients can be computed based on the spatial attenuation for elastic pressure and shear waves.

Without loss of generality, consider waves propagating along the x-axis. The waves are linear, so that small strains are assumed, see Ref. 131-132 for details.

The pressure waves are solutions of the form: $\mathbf{u} = [u(x), 0, 0]$. The corresponding momentum balance equation is

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{\partial s_{xx}}{\partial x}$$

where the stress component can be represented as

$$s_{xx} = \left(K + \frac{4}{3}G\right)\frac{\partial u}{\partial x} + \left(\eta_b + \frac{4}{3}\eta_v\right)\frac{\partial^2 u}{\partial t \partial x}$$

where K and G are the bulk and shear moduli, respectively; and η_b and η_v are the bulk and shear viscosity coefficients, respectively.

The shear waves are solutions of the form: $\mathbf{u} = [0, v(x), 0]$, where the y-polarization is assumed, again without loss of generality. The corresponding momentum balance equation is

$$\rho\frac{\partial^2 u}{\partial t^2} = \frac{\partial s_{xy}}{\partial x}$$

where

$$s_{xy} = G\frac{\partial v}{\partial x} + \eta_v\frac{\partial^2 v}{\partial t \partial x}$$

Consider the following generic form of the momentum balance as a representation of either case, pressure or shear waves:

$$\rho\frac{\partial^2 U}{\partial t^2} = D\frac{\partial^2 U}{\partial t^2} + \eta\frac{\partial^3 U}{\partial t \partial x^2}$$

In an infinite domain, the solution can be expressed for any specified angular frequency ω in the following complex-values form:

$$U(x,t) = U_+ \exp(i\omega t - ik_+(\omega)x) + U_- \exp(i\omega t - ik_-(\omega)x)$$

where U_+ and U_- are constant amplitudes, and

$$k_\pm = \pm(\kappa(\omega) - i\alpha(\omega))$$

are two solutions of the dispersion relation

$$\rho\omega^2 = Dk^2 + i\eta\omega k^2$$

Separation of the real and imaginary parts of the above equation gives

$$\kappa = \sqrt{\frac{\rho\omega^2\left(\sqrt{1+\left(\frac{\eta\omega}{D}\right)^2}+1\right)}{2D\left(1+\left(\frac{\eta\omega}{D}\right)^2\right)}} = \sqrt{\frac{\rho\omega^2}{D}}\left(1-\frac{3}{8}\left(\frac{\eta\omega}{D}\right)^2+O\left(\frac{\eta\omega}{D}\right)^4\right)$$

$$\alpha = \sqrt{\frac{\rho\omega^2\left(\sqrt{1+\left(\frac{\eta\omega}{D}\right)^2}-1\right)}{2D\left(1+\left(\frac{\eta\omega}{D}\right)^2\right)}} = \sqrt{\frac{\rho\omega^2}{D}}\left(\frac{1}{2}\frac{\eta\omega}{D}+O\left(\frac{\eta\omega}{D}\right)^3\right)$$

Note that for most materials, $\eta\omega \ll D$ even for very large values of ω.

The phase velocity is given by

$$c(\omega) = \frac{\omega}{\kappa} = \sqrt{\frac{2D\left(1+\left(\frac{\eta\omega}{D}\right)^2\right)}{\rho\left(\sqrt{1+\left(\frac{\eta\omega}{D}\right)^2}+1\right)}} = \sqrt{\frac{D}{\rho}}\left(1+\frac{3}{8}\left(\frac{\eta\omega}{D}\right)^2+O\left(\frac{\eta\omega}{D}\right)^4\right)$$

The solution takes the form

$$U(x,t) = U_+ e^{-\alpha(\omega)x} e^{-i\kappa(\omega)(x-c(\omega)t)} + U_- e^{\alpha(\omega)x} e^{i\kappa(\omega)(x+c(\omega)t)}$$

which can be seen as a superposition of two plane waves propagating in the opposite directions; both waves are spatially attenuated in the corresponding direction of propagation.

For each wave, the spatial attenuation is characterized by $\alpha(\omega)$ which is often called the attenuation coefficient. It has a SI unit (1/m), but is often given using neper per meter (Np/m). Alternatively, the value can be given in decibel per meter (dB/m). Both Np and dB are relative units (dimensionless). Values given in Np or Np/m can be used directly in any formula using SI units. The conversion formula needs to be applied for values using dB:

(value in Np) = (value in dB) $\ln(10)/20 = 0.1151$ (value in dB)

The wave attenuation can be also specified as attenuation per wavelength λ, which is defined as

$$\gamma(\omega) = \alpha(\omega)\lambda = 2\pi\frac{\alpha(\omega)}{\kappa(\omega)} = 2\pi\frac{\eta\omega/D}{\sqrt{1+\left(\frac{\eta\omega}{D}\right)^2}+1} = 2\pi\left(\frac{1}{2}\frac{\eta\omega}{D}+O\left(\frac{\eta\omega}{D}\right)^3\right)$$

which allows to find the viscosity as

$$\eta = \frac{2D}{\omega}\frac{\gamma}{2\pi}\left(1-\left(\frac{\gamma}{2\pi}\right)^2\right)^{-1}$$

The above formula can be directly applied to the cases of pressure and shear waves. Using given values of the attenuation per wavelength γ_p and γ_s measured in Np at the corresponding references frequencies $f_{p,ref}$ and $f_{s,ref}$, the shear and bulk viscosity are found as

$$\eta_v = \frac{2G}{\omega_{s,\,ref}}\frac{\gamma_s}{2\pi}\left(1-\left(\frac{\gamma_s}{2\pi}\right)^2\right)^{-1}$$

$$\eta_b = \frac{2\left(K+\frac{4}{3}G\right)}{\omega_{p,\,ref}}\frac{\gamma_p}{2\pi}\left(1-\left(\frac{\gamma_p}{2\pi}\right)^2\right)^{-1}-\frac{4}{3}\eta_v$$

where $\omega_{p,\,ref} = 2\pi f_{p,\,ref}$ and $\omega_{s,\,ref} = 2\pi f_{s,\,ref}$. If the values are given in dB per wavelength, the conversion factor $\ln(10)/20$ needs to be inserted.

For given values of attenuation coefficients α_p and α_s measured in Np/m at the corresponding references frequencies $f_{p,ref}$ and $f_{s,ref}$ the approximate formulas for the viscosity coefficients are

$$\eta_v = \frac{2Gc_s}{\omega_{s,\,ref}^2}\alpha_s$$

$$\eta_b = \frac{2\left(K+\frac{4}{3}G\right)c_p}{\omega_{p,\,ref}^2}\alpha_s - -\frac{4}{3}\eta_v$$

where the approximate phase velocity for the pressure and shear waves are given by $c_p = \sqrt{(K+4G/3)/\rho}$ and $c_s = \sqrt{G/\rho}$. If the values of α_p and α_s are given in dB/m, the conversion factor $\ln(10)/20$ needs to be inserted.

Loads and Constraints

In this section:

- Distributed Loads
- Gravity and Base Excitation
- Frame Acceleration Forces
- Spring Foundation and Thin Elastic Layer
- Added Mass
- Rigid Motion Suppression
- Inertia Relief
- Limited Displacement
- Low-Reflecting Boundary Condition
- Cyclic Symmetry and Floquet Periodic Conditions
- Symmetry Condition with Translation
- Roller Boundary Condition
- Thermal Expansion of Constraints
- Fluid–Structure Interaction
- Cell Periodicity
- Rigid Connector
- Continuity Condition

Distributed Loads

The direction of an explicitly applied distributed load must be given with reference to a local or global coordinate system in the spatial frame, but its magnitude must be with reference to the undeformed reference (or material) area. That is, the relation between the true force $\mathbf{f}$ acting on the current area da and the specified distributed load $\mathbf{F}$ acting on the material area dA is $\mathbf{f}da = \mathbf{F}dA$.

When the solid is subjected to an external pressure, p, the true force on a surface element acts with magnitude p in the current area da in the normal direction $\mathbf{n}$:

$$\mathbf{f} = p\mathbf{n}da$$

Therefore, the pressure load type specifies the distributed load as

$$\mathbf{F} = p\mathbf{n}\frac{da}{dA}$$

where both the normal $\mathbf{n}$ and area element da are functions of the current displacement field.

Another view of how to interpret the load, is to express it in the first Piola–Kirchhoff stress tensor P via the following formula:

$$\mathbf{F} = P \cdot \mathbf{n}_0$$

where the normal $\mathbf{n}_0$ corresponds to the undeformed surface element. Such a force vector is often referred to as the *nominal traction*.

Plane Stress

In a plane stress condition, the out-of-plane deformation causes the thickness to change, and this area effect is included explicitly.

The equation transforms to

$$\mathbf{F} = p\mathbf{n}\frac{dl}{dL}\left(1+\frac{\partial w}{\partial z}\right)$$

where l and L are the current and original in-plane lengths of the boundary.

Axial Symmetry

To account for the radial deformation changing the circumference and therefore the area element, the distributed load is applied as

$$\mathbf{F} = p\mathbf{n}\frac{da}{dA}\frac{(R+u)}{R}$$

Gravity and Base Excitation

Gravity is a global feature that applies forces to all features in the physics interface having a density, mass, or mass distribution. The gravitational force acts in a fixed spatial direction $\mathbf{e}_g$. Consider a domain feature having a density of ρ, the gravitational force is

$$\mathbf{g} = \rho g\mathbf{e}_g$$

where g is the acceleration of gravity. The default value of g is the earth's *standard acceleration of gravity*, having the value of `g_const` = 9.8066 m/s^2.

Base excitation is similar to gravity, but has two distinct properties: The sign is reversed (the acceleration of the foundation is given) and it defines variables for the global, room fixed, accelerations.

Frame Acceleration Forces

Add **Rotating Frame** and **Linearly Accelerated Frame** nodes to create the loads caused by the frame acceleration. This gives load contributions from all nodes in the physics interface that have a density or mass, such as **Linear Elastic Material**, **Rigid Material**, **Added Mass**, or **Point Mass**.

!	Only features which have a geometrical selection contribute to the mass forces. The **Mass and Moment of Inertia** nodes are global features and will not get any contribution from **Rotating Frame** and **Linearly Accelerated Frame** nodes.

In the following, the mass density ρ should be considered as generalized. It can represent mass per unit volume, mass per unit area, mass per unit length, or even mass, depending on the dimensionality of the object giving the contribution.

LINEARLY ACCELERATED FRAME

The force acts in a fixed spatial direction and is computed as

$$\mathbf{f} = -\rho \mathbf{a}_\mathrm{f}$$

where $\mathbf{a}_\mathrm{f}$ is a specified linear frame acceleration vector. Alternatively, the effective acceleration in the accelerated frame can be augmented to include the frame acceleration effect:

$$\mathbf{a}_\mathrm{tot} = \ddot{\mathbf{u}} + \mathbf{a}_\mathrm{f}$$

When computing the absolute velocity and displacement, the frame acceleration is integrated in time using the implicit Newmark–β method assuming a constant acceleration over the time step Δt,

$$\mathbf{v}_\mathrm{f}^{n+1} = \mathbf{v}_\mathrm{f}^{n} + \frac{1}{2}(\mathbf{a}_\mathrm{f}^{n+1} + \mathbf{a}_\mathrm{f}^{n})\Delta t$$

$$\mathbf{u}_{\mathrm{f}}^{n+1} = \mathbf{u}_{\mathrm{f}}^{n} + \mathbf{v}_{\mathrm{f}}^{n}\Delta t + \frac{1}{4}(\mathbf{a}_{\mathrm{f}}^{n+1} + \mathbf{a}_{\mathrm{f}}^{n})\Delta t^2$$

ROTATING FRAME

Centrifugal, Coriolis, and Euler forces are fictitious forces that need to be introduced in a rotating frame of reference, since it is not an inertial system. They can be added as loads. Alternatively, the effective acceleration in the rotating frame can be augmented to include the frame rotation effects:

$$\mathbf{a}_{\mathrm{tot}} = \ddot{\mathbf{u}} + \mathbf{a}_{\mathrm{f}}$$

where the rotating frame acceleration is computed as

$$\mathbf{a}_{\mathrm{f}} = \mathbf{a}_{\mathrm{cen}} + \mathbf{a}_{\mathrm{cor}} + \mathbf{a}_{\mathrm{eul}}$$

CENTRIFUGAL FORCE

A centrifugal force acts radially outward from the axis of rotation defined by the axial direction vector $\mathbf{e}_{\mathrm{ax}}$. The rotation is represented by the angular velocity vector

$$\Omega = \Omega\mathbf{e}_{\mathrm{ax}}$$

where Ω is the angular velocity. In vector form, the acceleration contribution and the loads are

$$\mathbf{a}_{\mathrm{cen}} = \Omega \times (\Omega \times \mathbf{r}_{\mathrm{p}})$$

$$\mathbf{F}_{\mathrm{cen}} = -\rho\mathbf{a}_{\mathrm{cen}}$$

where $\mathbf{r}_{\mathrm{p}}$ is the rotation position vector that contains the coordinates with respect to any point on the axis of rotation. The point is given by its radius vector in the global coordinate system $\mathbf{r}_{\mathrm{bp}}$.

SPIN-SOFTENING EFFECT

The structural displacement can be accounted for when computing the rotation position, so that

$$\mathbf{r}_{\mathrm{p}} = \mathbf{X} + \mathbf{u} - \mathbf{r}_{\mathrm{bp}}$$

This results in a contribution from the extra acceleration terms caused by the deformation into the system's stiffness matrix. The effect is often called spin-softening.

CORIOLIS FORCE

For a Coriolis force to appear, the object studied must have a velocity relative to the rotating frame. The acceleration contribution and the load are:

$$\mathbf{a}_{\mathrm{cor}} = 2\Omega \times \frac{\partial \mathbf{u}}{\partial t}$$

$$\mathbf{F}_{\mathrm{cor}} = -\rho \mathbf{a}_{\mathrm{cor}}$$

This gives a damping contribution since it is proportional to the velocity.

EULER FORCE

The Euler force occurs when the rate of rotation is not constant in time. The force acts in the plane of rotation perpendicular to the centrifugal force. The acceleration contribution and the load are

$$\mathbf{a}_{\mathrm{eul}} = \frac{\partial \Omega}{\partial t} \times \mathbf{r}_{\mathrm{p}}$$

$$\mathbf{F}_{\mathrm{eul}} = -\rho \mathbf{a}_{\mathrm{eul}}$$

Spring Foundation and Thin Elastic Layer

In this section, the equations for the spring type physics nodes are developed using boundaries, but the generalizations to geometrical objects of other dimensions are obvious. Also, for cases where rotational springs are present, the relations between moments and rotations are analogous to the relations between forces and displacements described below.

SPRING FOUNDATION

A spring gives a force that depends on the displacement and acts in the opposite direction. In the case of a force that is proportional to the displacement, this is called Hooke's law. In a suitable coordinate system, a spring condition can be represented as

$$\mathbf{f}_{\mathrm{s}} = -K \cdot (\mathbf{u} - \mathbf{u}_0)$$

where $\mathbf{f}_s$ is a force/unit area, $\mathbf{u}$ is the displacement deforming the spring, and K is a stiffness matrix. $\mathbf{u}_0$ is an optional deformation offset, which describes the stress-free state of the spring.

If the spring stiffness is not constant, then it is in general easier to directly describe the force as a function of the displacement, so that

$$\mathbf{f}_s = \mathbf{f}(\mathbf{u} - \mathbf{u}_0)$$

In the same way, a viscous damping can be described as a force proportional to the velocity

$$\mathbf{f}_v = -D(\dot{\mathbf{u}} - \dot{\mathbf{u}}_0)$$

where D is a matrix representing the viscosity.

Structural ("loss factor") damping is only relevant for frequency domain analysis and is defined as

$$\mathbf{f}_l = -i\eta K \cdot (\mathbf{u} - \mathbf{u}_0)$$

where η is the loss factor and i is the imaginary unit. It is also possible to give individual loss factors for each component in the stiffness matrix K.

If the elastic part of the spring definition is given as a force versus displacement relation, the stiffness K is taken as the stiffness at the linearization point at which the frequency response analysis is performed. Since the loss factor force is proportional to the elastic force, the equation can be written as

$$\mathbf{f}_l = i\eta \mathbf{f}_s$$

The contribution to the virtual work is

$$\delta W = \int_A (\mathbf{f}_s + \mathbf{f}_l + \mathbf{f}_v) \cdot \delta\mathbf{u} dA$$

THIN ELASTIC LAYER BETWEEN TWO PARTS

A spring or damper can also act between two boundaries of an identity pair. The spring force then depends on the difference in displacement between the two boundaries:

$$\mathbf{f}_{sD} = -\mathbf{f}_{sS} = -K(\mathbf{u}_D - \mathbf{u}_S - \mathbf{u}_0)$$

The uppercase indices refer to "source" and "destination". When a force versus displacement description is used,

$$\mathbf{f}_{sD} = -\mathbf{f}_{sS} = \mathbf{f}(\mathbf{u}-\mathbf{u}_0)$$
$$\mathbf{u} = \mathbf{u}_D - \mathbf{u}_S$$

The viscous and structural damping forces have analogous properties,

$$\mathbf{f}_{vD} = -\mathbf{f}_{vS} = -D(\dot{\mathbf{u}}_D - \dot{\mathbf{u}}_S - \dot{\mathbf{u}}_0)$$

$$\mathbf{f}_{lD} = -\mathbf{f}_{lS} = -i\eta K(\mathbf{u}_D - \mathbf{u}_S - \mathbf{u}_0)$$

or

$$\mathbf{f}_{lD} = i\eta\mathbf{f}_{sD}$$

The virtual work expression is formulated on the destination side of the pair as

$$\delta W = \int_{A_D} (\mathbf{f}_{sD} + \mathbf{f}_{lD} + \mathbf{f}_{vD}) \cdot \delta(\mathbf{u}_D - \mathbf{u}_S) dA_D$$

Here the displacements from the source side are obtained using the `src2dst` operator of the identity pair. Select the side with the finer mesh as destination if there is a difference in mesh density on the two sides of the pair.

THIN ELASTIC LAYER ON INTERIOR BOUNDARIES

On an interior boundary, the **Thin Elastic Layer** decouples the displacements between two sides of the boundary. The two boundaries are then connected by elastic and viscous forces with equal size but opposite directions, proportional to the relative displacements and velocities.

If an interface which is active on boundaries (Shell or Membrane in 3D for example) is added on the same interior boundary as a **Thin Elastic Layer**, then the virtual slit between the two sides of the boundary may be closed again. This happens if the domain interface and the boundary interface share the same displacement degrees of freedom.

The spring force can be written as

$$\mathbf{f}_{su} = -\mathbf{f}_{sd} = -K(\mathbf{u}_u - \mathbf{u}_d - \mathbf{u}_0)$$

or

$$\mathbf{f}_{su} = -\mathbf{f}_{sd} = \mathbf{f}(\mathbf{u}-\mathbf{u}_0)$$
$$\mathbf{u} = \mathbf{u}_u - \mathbf{u}_d$$

The viscous force is

$$\mathbf{f}_{vu} = -\mathbf{f}_{vd} = -D(\dot{\mathbf{u}}_u - \dot{\mathbf{u}}_d - \dot{\mathbf{u}}_0)$$

and the structural damping force is

$$\mathbf{f}_{lu} = -\mathbf{f}_{ld} = -i\eta K(\mathbf{u}_u - \mathbf{u}_d - \mathbf{u}_0)$$

or

$$\mathbf{f}_{lu} = i\eta\mathbf{f}_{su}$$

The subscripts u and d denote the "upside" and "downside" of the interior boundary, respectively.

The virtual work expression is formulated as

$$\delta W = \int_{A_u} (\mathbf{f}_{sd} + \mathbf{f}_{ld} + \mathbf{f}_{vd}) \cdot \delta(\mathbf{u}_u - \mathbf{u}_d) dA_u$$

STIFFNESS FROM MATERIAL DATA

When the stiffness is given in terms of actual material data and layer thickness d_s, the stiffness in the normal direction is computed based on a state of plane strain, so that

$$k_n = \frac{E(1-\nu)}{d_s(1+\nu)(1-2\nu)}$$

The assumption of plane strain conditions is relevant when the material in the elastic layer is softer than its surroundings, and this is normally the case.

The shear stiffness is isotropic in the tangential plane, having the value

$$k_t = \frac{G}{d_s} = \frac{E}{2d_s(1+\nu)}$$

Since the layer thickness is known in this case, it is also possible to compute a strain in the elastic layer. The strain tensor has the is stored in a variable with a name like `<`*`interface`*`>.<`*`feature`*`>.etel`*`ij`*, for example `solid.tel1.etelxx` for the normal strain. The two shear strains are stored in the `xy` and `xz` components of the tensor. In

3D, the orientation of the two local directions y and z used for the two shear strain directions is obtained using the following scheme:

1. Choose an auxiliary direction. Unless the normal to the layer is very close to the global X direction, use $\mathbf{e}_{\text{aux}} = \mathbf{e}_X$. If the X direction cannot be used, the Y direction is instead used as the auxiliary direction, $\mathbf{e}_{\text{aux}} = \mathbf{e}_Y$.
2. The local y direction is obtained from the part of the auxiliary direction which is orthogonal to the normal direction $\mathbf{n}$:

$$\mathbf{e}_y = \frac{\mathbf{e}_{\text{aux}} - (\mathbf{e}_{\text{aux}} \cdot \mathbf{n})\mathbf{n}}{|\mathbf{e}_{\text{aux}} - (\mathbf{e}_{\text{aux}} \cdot \mathbf{n})\mathbf{n}|}$$

3. The local z direction is orthogonal to the normal and the local y direction:

$$\mathbf{e}_z = \mathbf{n} \times \mathbf{e}_y$$

Springs and Dampers in the *Structural Mechanics Modeling* chapter.

Spring–Damper Theory

The **Spring–Damper** node is used to define elastic and dissipative forces between a source and a destination attachment. Two different spring types are available. When using the *directional* spring (Figure 3-39), all forces act in the direction of the spring extension. With the *matrix* spring formulation, more general dependencies between the displacements and forces can be defined.

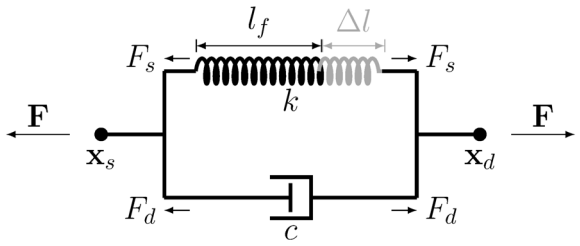


Figure 3-39: Conceptual sketch of a Spring–Damper.

- Connection Points and Spring Extension
- Deactivation

- Spring and Damping Forces
- Spring and Damping Energies

CONNECTION POINTS AND SPRING EXTENSION

The current position of the source and destination points, $\mathbf{x}_s$ and $\mathbf{x}_d$, can be written as

$$\mathbf{x}_s = \mathbf{X}_s + \mathbf{u}_s$$
$$\mathbf{x}_d = \mathbf{X}_d + \mathbf{u}_d$$

where $\mathbf{X}_s$ and $\mathbf{X}_d$ are the original positions of the two points, and $\mathbf{u}_s$ and $\mathbf{u}_d$ are their respective displacements. For the special case where the source point is selected as a fixed point,

$$\mathbf{x}_s = \mathbf{X}_s$$

The relative displacements between the source and destination points determines the forces in the spring–damper. Depending on the chosen spring type (directional or matrix), the relative displacement is defined by different quantities.

Directional

For the directional spring, the extension is given in terms of a scalar spring length. The initial spring length, l_0, is

$$l_0 = \|\mathbf{X}_d - \mathbf{X}_s\| = \sqrt{(\mathbf{X}_d - \mathbf{X}_s) \cdot (\mathbf{X}_d - \mathbf{X}_s)}$$

The current spring length, l, is

$$l = \|\mathbf{x}_d - \mathbf{x}_s\|$$

In the case of a geometrically linear analysis, the current spring length is linearized to

$$l = \frac{(\mathbf{x}_d - \mathbf{x}_s) \cdot (\mathbf{X}_d - \mathbf{X}_s)}{l_0}$$

Specify an initial spring extension Δl_0 in addition to the initial geometrical distance between the two points, so that the free length of the spring is

$$l_f = l_0 - \Delta l_0$$

It is also possible to specify the free length of the spring explicitly. The spring extension Δl is then computed as the difference between the current spring length and the free length l_f,

$$\Delta l = l - l_f$$

Matrix

When choosing the matrix spring type, the spring–damper forces are instead computed based on the relative displacement and the relative rotation vectors. If the study is geometrically linear, the relative rotation vector is

$$\Delta\Theta = \Theta_d - \Theta_s$$

where Θ_s and Θ_d are the rotations of the source and destination attachments, respectively. In case of geometrically nonlinear studies, the relative rotation vector is instead computed from the quaternion multiplication,

$$q_d = \Delta q \cdot q_s$$

where q_s and q_d are the source and destination rotations represented by unit quaternions, and Δq is a quaternion representing the differential rotation, which can be converted to an axis-angle representation defining the relative rotation vector.

$$\Delta\Theta = f(\Delta q)$$

By default, the relative displacement vector is simply defined as

$$\Delta\mathbf{u} = \mathbf{u}_d - \mathbf{u}_s$$

If the checkbox **Include rotational contribution in displacement** in the **Spring–Damper** section is selected, an additional term is included in the relative displacement vector which can be interpreted as the displacement at the tip of a bar element which connects the source and destination due to the rotation at the destination point. In a geometrically linear study, the relative displacement is then defined as

$$\Delta\mathbf{u} = \mathbf{u}_d - \mathbf{u}_s - \Theta_d \times (\mathbf{X}_d - \mathbf{X}_s)$$

In a geometrically nonlinear study, the cross product is replaced with the rotation matrix.

$$\Delta\mathbf{u} = \mathbf{u}_d - \mathbf{u}_s - (\mathbf{R}(\Theta_d) - \mathbf{I})(\mathbf{X}_d - \mathbf{X}_s)$$

DEACTIVATION

The spring–damper can be deactivated under certain conditions. In terms of the implementation, this means that many expressions are multiplied by an *activation*

indicator, i_{ac}. The activation indicator has the value 1 when the component is active, and 0 when deactivated.

SPRING AND DAMPING FORCES

Spring and damper forces are computed differently for the directional and the matrix spring types.

Directional

When the directional spring is chosen, the spring force is proportional to the spring constant k:

$$F_s = k\Delta l$$

If k depends on the extension, so that the spring is nonlinear, it should be interpreted as a secant stiffness, that is

$$F_s = k(\Delta l) \cdot \Delta l$$

Specify the spring force as function of extension explicitly as

$$F_s = F_s(\Delta l)$$

To create the expression for the function, use the built-in variable for the spring extension. It has the form `<physicsTag>.<SpringNodeTag>.dl` (for example, `solid.spd1.dl`).

In a dynamic analysis, the viscous damping force is computed as

$$F_d = c\frac{d}{dt}(\Delta l)$$

where c is the viscous damping coefficient.

The magnitude of the total force is

$$F = F_s + F_d$$

The total forces in the global coordinate system, acting on the destination and source points are

$$\mathbf{F}_{dst} = -\mathbf{F}_{src} = -F\frac{(\mathbf{x}_d - \mathbf{x}_s)}{l}$$

In a geometrically linear case, the orientation of the force is kept fixed, so that

$$\mathbf{F}_{\mathrm{dst}} = -\mathbf{F}_{\mathrm{src}} = -F\frac{(\mathbf{X}_{\mathrm{d}} - \mathbf{X}_{\mathrm{s}})}{l}$$

The contribution to the virtual work is

$$\mathbf{F}_{\mathrm{dst}} \cdot (\delta\mathbf{u}_{\mathrm{d}} - \delta\mathbf{u}_{\mathrm{s}})$$

Matrix

When choosing the *matrix* formulation, the spring and the damper forces are computed from the relative displacements and rotations. In the most general case, the spring–damper force, $\mathbf{F}$, and the spring–damper moment, $\mathbf{M}$, are

$$\begin{bmatrix}\mathbf{F}\\ \mathbf{M}\end{bmatrix} = \begin{bmatrix}\mathbf{F}_{\mathrm{s}}\\ \mathbf{M}_{\mathrm{s}}\end{bmatrix} + \begin{bmatrix}\mathbf{F}_{\mathrm{d}}\\ \mathbf{M}_{\mathrm{d}}\end{bmatrix} = \begin{bmatrix}\mathbf{k}_{\mathrm{u}} & \mathbf{k}_{\mathrm{u}\Theta}\\ \mathbf{k}_{\Theta\mathrm{u}} & \mathbf{k}_{\Theta}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{u}\\ \Delta\Theta\end{bmatrix} + \begin{bmatrix}\mathbf{c}_{\mathrm{u}} & \mathbf{c}_{\mathrm{u}\Theta}\\ \mathbf{c}_{\Theta\mathrm{u}} & \mathbf{c}_{\Theta}\end{bmatrix}\begin{bmatrix}\Delta\dot{\mathbf{u}}\\ \Delta\dot{\Theta}\end{bmatrix}$$

where the subscripts *s* and *d* refer to the spring and damper, respectively; the different $\mathbf{k}$ and $\mathbf{c}$ represent submatrices of size 3-by-3, describing the elastic and the dissipative forces and moments. In 2D, some of the matrices related to the rotation and to the translational-rotational coupling have components which are zero by definition.

SPRING AND DAMPING ENERGIES

In stationary and time-dependent analysis, the elastic energy in the spring is computed.

Directional

With the directional spring formulation, the energy is

$$W_{\mathrm{s}} = i_{\mathrm{ac}}\int_0^{\Delta l} F_{\mathrm{s}}\,d(\Delta l)$$

In a time-dependent analysis, the energy dissipated in the damper, W_{d}, is computed using an extra degree of freedom. The following equation is added:

$$\frac{dW_{\mathrm{d}}}{dt} = i_{\mathrm{ac}}F_{\mathrm{d}}\frac{d(\Delta l)}{dt}$$

In a frequency domain analysis, the elastic energy in the spring and the energy dissipated in the damper are computed as

$$W_s = i_{ac}\frac{1}{4}\mathrm{Re}(\mathrm{lindev}(F_s)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

$$W_d = i_{ac}\frac{1}{4}\mathrm{Re}(\mathrm{lindev}(F_d)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

These energy quantities represent the cycle average, and only the perturbation terms are included.

Matrix

If instead the *matrix* formulation is used, the elastic energy in the spring is

$$W_s = i_{ac}\frac{1}{2}(\mathbf{F}_s\cdot\Delta\mathbf{u} + \mathbf{M}_s\Delta\Theta)$$

In a time-dependent analysis, the energy dissipated in the damper, W_d, is computed using an extra degree of freedom. The following equation is added:

$$\frac{dW_d}{dt} = i_{ac}(\mathbf{F}_s\cdot\Delta\dot{\mathbf{u}} + \mathbf{M}_s\Delta\dot{\Theta})$$

In a frequency domain analysis, the elastic energy in the spring and the energy dissipated in the damper are computed as

$$W_s = i_{ac}\frac{1}{4}[\mathrm{Re}(\mathrm{lindev}(\mathbf{F}_s)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta\mathbf{u}))) +$$
$$\mathrm{Re}(\mathrm{lindev}(\mathbf{M}_s)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta\Theta)))]$$

$$W_d = i_{ac}\frac{1}{4}[\mathrm{Relindev}(\mathbf{F}_d)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta\mathbf{u}))) +$$
$$\mathrm{Re}(\mathrm{lindev}(\mathbf{M}_d)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta\Theta)))]$$

Added Mass

The Added Mass node can be used for supplying inertia that is not part of the material itself. Such inertia does not need to be isotropic, in the sense that the inertial effects are not the same in all directions. This is, for example, the case when a structure immersed in a fluid vibrates. The fluid is added to the inertia for acceleration in the direction normal to the boundary, but not tangential to it.

Other uses for added mass are when sheets or strips of a material that is heavy, but having a comparatively low stiffness, are added to a structure. The data for the base

material can then be kept unaltered, while the added material is represented purely as added mass.

The value of an added mass can also be negative. Use such a negative value for adjusting the mass when a part imported from a CAD system does not get exactly the correct total mass due to simplifications of the geometry.

Added mass can exist on domains, boundaries, and edges. The inertial forces from added mass can be written as

$$\mathbf{f}_{\mathrm{m}} = -M\frac{\partial^2 \mathbf{u}}{\partial t^2}$$

where M is a diagonal mass distribution matrix. For added mass on a boundary (and for objects of other dimensions), the contribution to the virtual work is

$$\delta W = \int_A \mathbf{f}_{\mathrm{m}} \cdot \delta \mathbf{u} dA$$

Rigid Motion Suppression

The Rigid Motion Suppression feature is a convenient way to automatically create a set of constraints which are sufficient to inhibit rigid body modes. The constraints are selected so that no reaction forces are introduced as long as the external loads are in equilibrium.

Consider that the undeformed and deformed positions of a body are measured in the same coordinate system, see Lagrangian Formulation for details. These are related by the displacement vector as

$$\mathbf{x} = \mathbf{X} + \mathbf{u}(\mathbf{X}, t)$$

where $\mathbf{x}$ is the *spatial coordinate*, $\mathbf{X}$ is the *material coordinate*, and $\mathbf{u}$ is the displacement.

An average displacement of the body can be obtained by a volume average, which requires a domain integration. However, a computationally leaner operation is to only consider the domain boundaries or geometrical points when averaging the displacement. When defining the rigid motion constraints, it is sufficient to only consider the geometric points of the body.

Constraining the average displacement to zero is equivalent to having the same average position (barycenter or centroid) in the undeformed and deformed body. Since the average displacement can be written as

$$\bar{\mathbf{x}} = \bar{\mathbf{X}} + \bar{\mathbf{u}}$$

The constraint

$$\bar{\mathbf{u}}(\mathbf{X}, t) = \mathbf{0}$$

is imposed, so the average centroid is the same for the deformed and undeformed body. This is enforced by adding three displacement constraints in 3D or two constraints in 2D.

The averaging is done by considering all the geometrical points in the domain, but it is also possible to consider a subset of points.

Three additional constraints in 3D, or one additional constraint in 2D are required to constrain the rotation with respect to the average centroid position $\bar{\mathbf{X}}$

$$(\mathbf{X} - \bar{\mathbf{X}}) \times \mathbf{u}$$

Inertia Relief

Inertia relief analysis is a type of stationary analysis, where the external loads are balanced by (unknown) inertial forces. Thus, even though rigid body motion constraints must be present to avoid a singular stiffness matrix, there will be no reaction forces.

In a 3D geometry, three selected control points are given by their radius-vectors $\mathbf{r}_j$. Using these points, a local coordinate system is defined with the origin at point 1 and base vectors given by

$$\mathbf{e}_1 = (\mathbf{r}_2 - \mathbf{r}_1) / |\mathbf{r}_2 - \mathbf{r}_1|$$

$$\mathbf{e}_2 = (\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1) / |(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)|$$

$$\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$$

The transform from the global to local system is given by

$$\mathbf{T} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$$

The displacement vectors in the local system at point (j) are computed as

$$\{u, v, w\}_j = \mathbf{T}\mathbf{u}_j$$

where $\mathbf{u}_j$ are displacement vector DOF at the corresponding point.

The radius vector at the local system is given by

$$\mathbf{r}_\mathrm{b} = \mathrm{T}(\mathbf{r}_1 - \{X, Y, Z\})$$

and the frame acceleration field in the global system is computed as

$$\mathbf{a}_\mathrm{f} = \mathrm{T}^T(\mathbf{a} + \mathbf{o} \times \mathbf{r}_\mathrm{b})$$

Here, $\mathbf{a}$ is the translational acceleration vector and $\mathbf{o}$ is the angular acceleration vector.

During the *training study* step, unit linear or angular accelerations are applied as

$$\mathbf{a} = \{a_1, a_2, a_3\} = \{if(\texttt{loadcase1}), if(\texttt{loadcase2}), if(\texttt{loadcase3})\}$$

$$\mathbf{o} = \{o_1, o_2, o_3\} = \{if(\texttt{loadcase4}), if(\texttt{loadcase5}), if(\texttt{loadcase6})\}$$

In the seventh and last load case, both $\mathbf{a}$ and $\mathbf{o}$ are set to zero. Thus, the external load is present in all load cases.

Apply the following six weak point constraints:

$$\{u, v, w\}_1 = 0$$

at point 1, which constrains translations in directions $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}3$. The corresponding three scalar constraints are numbered as 1,2,3. At point 2,

$$\{v, w\}_2 = 0$$

which constrains rotations around axes $\mathbf{e}_3$ and $\mathbf{e}_2$. The corresponding two scalar constraints are numbered as 4,5. Finally,

$$\{v\}_3 = 0$$

at point 3, which constrains rotation around axis $\mathbf{e}_1$. This scalar constraint is numbered as 6.

The corresponding constraint force components for the six load cases are represented by a 6x6 matrix

$$\mathrm{f} = [f_{mn}]$$

where the first index (m) is the number of the constraint, and the second index (n) is the number of the load case. For load case 7, the constraint force components are represented by a 6-component vector

$$\mathbf{F} = \{F_m\}$$

For any other study than the training study, the frame acceleration is computed using

$$\{a_1, a_2, a_3, o_1, o_2, o_3\} = (\mathbf{F} \otimes \{1, 1, 1, 1, 1, 1\} - \mathrm{f})^{-1}\mathbf{F}$$

with the values of f and $\mathbf{F}$ precomputed during the training study. Applying the corresponding frame acceleration in the global system will result into zero constraint forces for all six constraints.

In the case of 2D geometry, only the constraints 1,2 and 4 are needed. The corresponding load cases are 1,2 and 6. Thus, f can be reduced to a 3x3 matrix, and $\mathbf{F}$ will be a 3-component vector.

In 2D axial symmetry, only constraint 3 and load cased 3 are needed.

Limited Displacement

In some physics interfaces, displacements can be prescribed not only to a certain value, but also to move freely within given limits. This is specified using the **Limited** option in the **Prescribed Displacement** or **Prescribed Displacement/Rotation** nodes.

The limit conditions are implemented as a weak inequality constraint. For each displacement component u_i, a gap distance is computed as

$$g_i = \min(-u_i + u_{0i,\max}, u_i - u_{0i,\min}) \tag{3-199}$$

where $u_{0i,\max}$ and $u_{0i,\min}$ are the maximum and minimum limits. Given Equation 3-199, the weak inequality constraint is formulated as

$$0 = \int f_i \delta g_i dX \tag{3-200}$$

and is subjected to the Kuhn–Tucker conditions

$$g_i \geq 0, \quad f_i \geq 0, \quad f_i g_i = 0$$

where f_i is the constraint force (or contact reaction force). The integral is taken over the geometric entities of the selection, which may be domains, boundaries, edges, or points. For a point selection, the integral should be interpreted as a summation. The constraint force will have corresponding units.

Using the penalty method to regularize the constraint, the contact reaction force is defined as

$$f_i = \max(-k_p g_i, 0)$$

where k_p is the penalty factor. With the above definition of f_i, Equation 3-200 is added as a weak contribution to the model to implement the constraint. The unit of the penalty factor depends on the type of geometrical entity that is constrained.

The augmented Lagrangian implementation of the inequality constraint is based on the following augmentation of Equation 3-200

$$\begin{aligned} 0 &= \int f_i \delta g_i dX \\ 0 &= -\frac{1}{k_p}\int (f_i - f_{\mathrm{lm},i})\delta f_{\mathrm{lm},i} dX \end{aligned} \qquad (3\text{-}201)$$

where $f_{\mathrm{lm},i}$ is a Lagrange multiplier that is added as an extra degree of freedom to the model. The penalized contact reaction force f_i is for the augmented Lagrangian method defined as

$$f_i = \max(-k_p g_i + f_{\mathrm{lm},i}, 0)$$

and the constraint is implemented by adding the weak contributions defined in Equation 3-201 to the model.

Low-Reflecting Boundary Condition

The low-reflecting boundary condition is mainly intended for letting waves pass out from the model domain without reflection in time-dependent analyses. It is also available in the frequency domain, but then adding a perfectly matched layer (PML) is usually a better option.

As a default, the low-reflecting boundary condition takes the material data from the adjacent domain in an attempt to create a perfect impedance match for both pressure waves and shear waves, so that

$$\sigma \cdot \mathbf{n} = -\rho c_p \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n}\right)\mathbf{n} - \rho c_s \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{t}\right)\mathbf{t}$$

where $\mathbf{n}$ and $\mathbf{t}$ are the unit normal and tangential vectors at the boundary, respectively, and c_p and c_s are the speeds of the pressure and shear waves in the material. This approach works best when the wave direction in close to the normal at the wall.

In the general case use

$$\sigma \cdot \mathbf{n} = -\mathbf{d}_{im}(\rho, c_p, c_s)\frac{\partial \mathbf{u}}{\partial t}$$

where the mechanical impedance $\mathbf{d}_{im}$ is a diagonal matrix available as the user input, and by default it is set to

$$\mathbf{d}_{im} = \rho \frac{c_p + c_s}{2}\mathbf{I}$$

More information about modeling using low-reflecting boundary conditions can be found in Ref. 157.

Cyclic Symmetry and Floquet Periodic Conditions

These boundary conditions are based on the Floquet theory which can be applied to the problem of small-amplitude vibrations of spatially periodic structures. It is also applicable when performing a linear buckling analysis.

If the problem is to determine the frequency response to a small-amplitude time-periodic excitation that also possesses spatial periodicity, the theory states that the solution can be sought in the form of a product of two functions. One follows the periodicity of the structure, while the other one follows the periodicity of the excitation. The problem can be solved on a unit cell of periodicity by applying the corresponding periodicity conditions to each of the two components in the product.

This section describes the theory for solids in 3D, but is equally applicable to shells. In the case of shells, the periodicity condition is applied to edges, and the rotational degrees of freedom (displacements of the normal vector) are treated in the same way as the translational displacements.

The problem can be modeled using the full solution without applying the above described multiplicative decomposition. For such a solution, the Floquet periodicity conditions at the corresponding boundaries of the periodicity cell are expressed as

$$\mathbf{u}_{\text{destination}} = \exp[-i\mathbf{k}_F(\mathbf{r}_{\text{destination}} - \mathbf{r}_{\text{source}})]\mathbf{u}_{\text{source}}$$

where $\mathbf{u}$ is a vector of dependent variables, $\mathbf{r}$ is the position, and the vector $\mathbf{k}_F$ represents the spatial periodicity of the excitation.

The cyclic symmetry boundary condition presents a special but important case of Floquet periodicity, for which the unit periodicity cell is a sector of a structure that consists of a number of identical sectors. The frequency response problem can then be solved in one sector of periodicity by applying the periodicity condition. The situation is often referred to as *dynamic cyclic symmetry*.

For an eigenfrequency study, all the eigenmodes of the full problem can be found by performing the analysis on one sector of symmetry only and imposing the cyclic symmetry of the eigenmodes with an angle of periodicity $\varphi = m\theta$, where the cyclic symmetry mode number m can vary from 0 to $N/2$, with N being the total number of sectors so that $\theta = 2\pi/N$.

The Floquet periodicity conditions at the sides of the sector of symmetry can be expressed as

$$\mathbf{u}_{\text{destination}} = e^{-i\varphi}R_\theta\mathbf{u}^T_{\text{source}}$$

where the $\mathbf{u}$ represents the displacement vectors with the components given in the default Cartesian coordinates. Multiplication by the rotation matrix given by

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

makes the corresponding displacement components in the cylindrical coordinate system differ by the factor $\exp(-i\varphi)$ only. For scalar dependent variables, a similar condition applies, for which the rotation matrix is replaced by a unit matrix.

The angle φ represents either the periodicity of the eigenmode for an eigenfrequency analysis or the periodicity of the excitation signal in case of a frequency-response analysis. In the latter case, the excitation is typically given as a load vector

$$\mathbf{F} = -\mathbf{F}_0\exp[-im\,\text{atan}(Y/X)]$$

when modeled using the Cartesian coordinates. The parameter m is often referred to as the *azimuthal wave number*.

When using cyclic symmetry, you may get computed modes having a mode number other than the one you asked for. Such modes are also correct solutions. This phenomenon is an effect of the fact that the trigonometric boundary conditions are only unique up to a factor 2π. The general rule is that if you search for modes with mode number m, you can get any mode with mode number $|m + kN|$. Here, N is the number of sectors for a full revolution, and k is any integer, positive or negative.

If your model is fully rotationally symmetric, then it is a good idea to use a very small sector in the analysis. Not only will you have a low number of degrees of freedom, but since N is then large, you will usually get the modes you asked for. Modes with high mode numbers have complicated patterns that will cause high natural frequencies (or, in the case of buckling, high critical load factors).

Ref. 158 contains more information about cyclic symmetry conditions.

Symmetry Condition with Translation

A standard symmetry condition prohibits translation in the symmetry direction.

$$\mathbf{u} \cdot \mathbf{n}_s = 0$$

The normal to the symmetry plane, $\mathbf{n}_s$, is either computed from the geometry, or given as a user input.

When allowing translation of the symmetry plane, the constraint is modified to

$$\mathbf{u} \cdot \mathbf{n}_s - u_n = 0$$

Here, u_n is a single scalar displacement offset which is the same for all locations where the symmetry constraint is applied.

If the selected normal direction condition is **Prescribed displacement**, then u_n is the given value.

If the selected normal direction condition is **Free displacement** or **Prescribed force**, then u_n is added as a degree of freedom to be determined during the solution.

The weak contribution for this equation is

$$F_n \text{test}(u_n)$$

where F_n is the prescribed force. Note that in the **Free displacement** case, the force is zero, so no weak contribution is added. This corresponds to a solution with zero reaction force.

Roller Boundary Condition

The **Roller** boundary condition is similar to a **Symmetry** boundary condition, since it constrains the displacement in a direction normal to the boundary. A **Roller** is however intended to be used also on curved boundaries. The constraint can be formulated as

$$\mathbf{u} \cdot \mathbf{n}_r = 0$$

where the normal $\mathbf{n}_r$ is computed using different methods depending on the selection in the **Roller Constraint** section.

When **Normal orientation** is set to **Automatic**, the normal orientation is computed from the mesh, or its underlying geometry. Consider for example a roller condition on a planar surface. Theoretically, the normal at all node points should be parallel. But if there are inaccuracies in the node locations, the computed normals may not be exactly the same everywhere. They can differ not only between nodes, but also between neighboring elements connected to the same node. Thus, there may be constraints acting in somewhat different directions. Such constraints can make the boundary appear as fixed, rather than sliding. This potential problem can be reduced if you select **Nodal** as **Constraint method** in the **Constraint Setting** section.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Settings

In the modeling section of the *Structural Mechanics User's Guide*:

- Elemental and Nodal Constraints

If you select one of the explicit shapes (**Plane**, **Cylinder**, or **Sphere**) as **Normal orientation**, then the orientation of the normal $\mathbf{n}_r$ is instead a user input, so there will be no problem with numerical inaccuracies.

If you select **Plane**, you give the direction explicitly as a constant vector.

If you select **Cylinder**, the normal is computed as being perpendicular to the cylinder axis at each node point. The input defining the cylinder is a **Point on axis**, $\mathbf{X}_c$, and the vector along the cylinder axis, $\mathbf{e}_s$. For a node point located at an original coordinate $\mathbf{X}$, the normal orientation is computed as

$$\mathbf{X}' = \mathbf{X} - \mathbf{X}_c$$
$$\mathbf{n}_r = \mathbf{X}' - (\mathbf{X}' \cdot \mathbf{e}_s)\mathbf{e}_s$$

Here it is not necessary to normalize $\mathbf{n}_r$. It is actually the radial vector from the cylinder axis to the location $\mathbf{X}$.

In the case of geometric nonlinearity, the orientation of the normal would change. This is implemented as a nonlinear constraint, where the node is forced to maintain its distance from the cylinder axis, while allowed to move freely in the axial and circumferential directions. Thus, the normal orientation is not explicitly computed.

The constraint expression is

$$(|\mathbf{x}'|^2 - (\mathbf{x}' \cdot \mathbf{e}_s)^2) - (|\mathbf{X}'|^2 - (\mathbf{X}' \cdot \mathbf{e}_s)^2) = 0$$

where

$$\mathbf{X}' = \mathbf{X} - \mathbf{X}_c$$
$$\mathbf{x}' = \mathbf{x} - \mathbf{X}_c$$

The radius of the cylinder is not given explicitly, so each node will maintain its own original distance from the cylinder axis.

If you select **Sphere**, the normal is computed as the direction from the center of the sphere to each node point. The only input is a **Center of sphere**, $\mathbf{X}_c$. For a node point located at an original coordinate $\mathbf{X}$, the normal (actually, the radial vector) is computed as

$$\mathbf{n}_r = \mathbf{X} - \mathbf{X}_c$$

In the case of geometric nonlinearity, the orientation of the normal would change. This is implemented as a nonlinear constraint, where the node is forced to maintain its distance from the center of the sphere, while allowed to move freely in the two directions on the sphere surface. Thus, the normal orientation is not explicitly computed. The constraint is written as

$$|\mathbf{x} - \mathbf{X}_c|^2 - |\mathbf{X} - \mathbf{X}_c|^2 = 0$$

The radius of the sphere is not given explicitly, so each node will maintain its own original distance from the center.

Thermal Expansion of Constraints

Constraints like **Fixed Constraint** and **Prescribed Displacement** will in general cause stresses near the constrained boundaries when the structure undergoes temperature changes. The same is true also for rigid objects like **Rigid Material**, **Rigid Connector**, and **Attachment**. By adding a **Thermal Expansion** subnode to these features, you can allow the constrained boundaries to have a thermal expansion displacement.

The thermal strains will in general have a spatial distribution given by

$$\varepsilon(\mathbf{X}) = \alpha(\mathbf{X}, T)(T - T_{\text{ref}})$$

Note that this is the thermal expansion of the virtual surroundings of the structure being analyzed, so it is unrelated to the thermal strains of the structure itself.

The strain field must be converted into a displacement field $\mathbf{u}(\mathbf{X})$ such that

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial x_i}\right)$$

If the strain field fulfills the general compatibility relations, it is in principle possible to integrate the above relation. The procedure is outlined in Ref. 159, giving

$$u_i(\mathbf{X}) = u_i(\mathbf{X}_0) + \int_{\mathbf{X}_0}^{\mathbf{X}} \left(\varepsilon_{il} + (X_k - X'_k)\left(\frac{\partial \varepsilon_{il}}{\partial X_k} - \frac{\partial \varepsilon_{kl}}{\partial X_i}\right)\right) dX'_l$$

Summation over double indices is implied. The rigid body rotation term given in Ref. 159 is omitted, since it cannot be derived from the strain field. The reference point is chosen so that the displacement (caused by the strain field) is zero, so that the $u_i(\mathbf{x}_0)$ term can be omitted. The integral is path independent when the compatibility is fulfilled. Because the constrained region is a virtual object, the integration path does

not have to be inside a domain. For simplicity, a straight line from $\mathbf{X}_0$ to $\mathbf{X}$ is used for the integration. Let $\mathbf{p}$ be the vector between the two points,

$$\mathbf{p} = \mathbf{X} - \mathbf{X}_0$$

The distance along the integration path can then be parameterized by a parameter s running from 0 to 1as

$$\mathbf{X}' = \mathbf{X}_0 + s\mathbf{p}$$

giving

$$u_i(\mathbf{X}) = \int_0^1 \left(\varepsilon_{il} + (1-s)p_k\left(\frac{\partial \varepsilon_{il}}{\partial X_k} - \frac{\partial \varepsilon_{kl}}{\partial X_i}\right)\right)p_l ds \tag{3-202}$$

This integral can be computed using the built-in `integrate()` operator as long as the strain field is an explicit function of the material frame coordinates $\mathbf{X}$.

For the physics interfaces which have rotational degrees of freedom (Beam, Shell, and Plate), not only the displacement, but also the rotation of the constraint is needed. For a given displacement field $\mathbf{u}(\mathbf{X})$, the infinitesimal rotation vector Θ is given by

$$\Theta = \frac{1}{2}\nabla \times \mathbf{u}$$

Applying the rotation operator to Equation 3-202 gives

$$\Theta_m(\mathbf{X}) = \frac{1}{2}\varepsilon_{mni}\int_0^1 \left((3-2s)\frac{\partial \varepsilon_{il}}{\partial X_n} + (1-s)p_k\frac{\partial^2 \varepsilon_{il}}{\partial X_k \partial X_n}\right)p_l ds$$

where ε_{mni} is the permutation tensor.

Note that the general compatibility requirements will not be fulfilled for arbitrary expressions for the thermal strain distribution. In such cases, the stresses caused by the constraints cannot completely be removed, but they will be significantly decreased. The results will then depend on the choice of reference point.

Constraints and Thermal Expansion in the *Structural Mechanics Modeling* chapter.

Fluid–Structure Interaction

Fluid–Structure Interaction multiphysics coupling combines fluid flow with structural mechanics to capture the interaction between the fluid and the solid structure. A structural mechanics interface (Solid Mechanics, Shell, Membrane, or Multibody Dynamics) and a Single-Phase Flow interface model the solid and the fluid, respectively. The Fluid–Structure Interaction (FSI) couplings appear on the boundaries between the fluid and the solid. The physics interface uses an arbitrary Lagrangian–Eulerian (ALE) method to combine the fluid flow formulated using an Eulerian description and a spatial frame with solid mechanics formulated using a Lagrangian description and a material (reference) frame.

The fluid flow is described by the Navier–Stokes equations, which provide a solution for the velocity field $\mathbf{u}_{\text{fluid}}$. The total force exerted on the solid boundary by the fluid is the negative of the reaction force on the fluid,

$$\mathbf{f} = \mathbf{n} \cdot \left\{ -p\mathbf{I} + \left(\mu(\nabla \mathbf{u}_{\text{fluid}} + (\nabla \mathbf{u}_{\text{fluid}})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u}_{\text{fluid}})\mathbf{I} \right) \right\} \tag{3-203}$$

where p denotes pressure, μ the dynamic viscosity for the fluid, $\mathbf{n}$ the outward normal to the structure, and $\mathbf{I}$ the identity matrix. Because the Navier–Stokes equations are solved in the spatial (deformed) frame while the structural mechanics interfaces are defined in the material (undeformed) frame, a transformation of the force is necessary. This is done according to

$$\mathbf{F} = \mathbf{f} \cdot \frac{dv}{dV}$$

where dv and dV are the mesh element scale factors for the spatial frame and the material (reference) frame, respectively.

The coupling in the other direction consists of the structural velocity

$$\frac{\partial \mathbf{u}_{\text{struct}}}{\partial t}$$

(the rate of change for the displacement of the structure), which acts as a moving wall for the fluid domain.

The structural mechanics formulations support geometric nonlinearity (large deformations). The spatial frame also deforms with a mesh deformation that is equal to the displacements $\mathbf{u}_{\text{struct}}$ of the solid within the solid domains. The mesh is free to

move inside the fluid domains, and it adjusts to the motion of the solid walls. This geometric change of the fluid domain is automatically accounted for in COMSOL Multiphysics by the ALE method.

ONE-WAY FLUID–STRUCTURE INTERACTION

For small values of the structural displacement and its rate of change, the Fluid–Structure Interaction interface includes one-way model formulations. The one-way models sequentially solve for the fluid flow, compute the load from Equation 3-203, and then apply it in the solution for the solid displacement. Since these methods are unidirectional, the mesh deformation is excluded from the solution. When applicable, the one-way versions offer computationally cheaper alternatives to a fully coupled counterpart.

Stationary and Time Dependent one-way studies are available for selection from the Preset Studies branch when adding a study. These studies include a **Fluid** study step and a **Solid** study step. When an additional physics interface is added to the model, it is by default added to both study steps.

In this case the one-way study steps display under **Preset Studies for Some Physics Interfaces** branch since the Fluid study step does not solve for the solid displacement and vice versa. When using a turbulence model requiring the distance to the closest wall, the Preset Studies includes a **Wall Distance Initialization** study step.

When solving a transient one-way FSI model, besides saving the solution from the Fluid study step with adequate frequency, it is advisable to save the solution from the Solid study step at the same times as the fluid solution. This way, all the information from the Fluid study step is used in the Solid study step.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

Cell Periodicity

The heterogeneous properties of a material at the microscopic scale are often impractical for direct application at the macroscopic scale, which is typically the focus of structural analysis. In such cases, it becomes necessary to employ a homogeneous material model that incorporates appropriately averaged properties.

The Cell Periodicity feature facilitates the evaluation of such average properties. It is based on the idea of a *repeating unit cell* (RUC) or a *representative volume element*

(RVE). The cell is a microscopic domain that is representative for the material on a macroscopic length scale.

The terms RVE (Representative Volume Element) and RUC (Representative Unit Cell) are frequently used interchangeably in the academic literature. Generally, an RVE describes a subvolume that characterizes heterogeneous materials with a statistically homogeneous microstructure. Conversely, the RUC is employed to represent subvolumes characterized by periodic patterns.

When dealing with materials characterized by a random distribution of particles, fibers, or pores, the appropriate subvolume to consider would be an RVE. To be numerically efficient, this RVE should be as small as possible, but sufficiently large to provide a representative sample for the determination of average material properties at a macroscopic scale.

If the material or lattice exhibits true periodicity, the suitable choice for representing a material subvolume would be an RUC, which repeats itself to uniformly fill the space with a repetitive pattern.

The distinction between RVEs and RUCs necessitates the application of different sets of boundary conditions. The **Cell Periodicity** node offers two distinct sets of boundary conditions, namely **Periodic** and **Homogeneous**, designed to suit these subvolumes.

To model a microscopic structure, you add a **Cell Periodicity** node, and select the domains representing the unit cell. For each pair of matching boundaries, add a **Boundary Pair** subnode, and select the boundaries.

There are no limitations on the physics features you can use for modeling the cell, as long as the basic assumptions about homogenization are respected. You should however not add any displacement constraints, since the possible rigid body motions are automatically constrained by the **Cell Periodicity** node.

HOMOGENIZATION METHOD

The homogenization method introduces two scales: a macroscale and a microscale. The macroscale usually refers to the homogenized continuous medium, and the microscale to the heterogeneous cell. The macroscopic stress tensor σ_{avg} and the macroscopic strain tensor ε_{avg} are derived by averaging the stresses and strains in the cell

$$\sigma_{avg} = \frac{1}{V}\int_V \sigma dV \text{ and } \varepsilon_{avg} = \frac{1}{V}\int_V \varepsilon dV \tag{3-204}$$

where V is the volume of the cell. The macroscopic elasticity tensor $\bar{C}$ of the homogenized continuum is then defined by

$$\sigma_{avg} = \bar{C}:\varepsilon_{avg} \tag{3-205}$$

PERIODIC BOUNDARY CONDITIONS

For a periodic structure that consists of an array of RUC, the displacement field is written as

$$\mathbf{u} = \varepsilon_{avg}\mathbf{X} + \mathbf{u}^*$$

where ε_{avg} is the global average strain tensor, $\mathbf{X}$ is the position, and $\mathbf{u}^*$ is a function that is periodic from one cell to another. Given that the array of cells forms a continuous structure, it is essential to ensure continuity of displacements across the boundaries between these cells. The boundaries in a **Boundary Pair** subnode must therefore always appear in parallel pairs. One of them is labeled as *source* and the other as *destination*. The displacements on a pair of parallel (and opposite) boundaries can be written as

$$\mathbf{u}_{dst} = \varepsilon_{avg}\mathbf{X}_{dst} + \mathbf{u}^* \text{ and } \mathbf{u}_{src} = \varepsilon_{avg}\mathbf{X}_{src} + \mathbf{u}^*$$

where $\mathbf{X}_{dst}$ and $\mathbf{u}_{dst}$ are the position and displacement on the destination side, and $\mathbf{X}_{src}$ and $\mathbf{u}_{src}$ are the position and displacement on the source side.

Hence, the displacement boundary condition is established by

$$\mathbf{u}_{dst} - \mathbf{u}_{src} = \varepsilon_{avg}(\mathbf{X}_{dst} - \mathbf{X}_{src}) \tag{3-206}$$

The continuity of tractions should also be satisfied for RVEs together with the continuity of displacements. This means that periodic conditions for tractions are equivalent to the periodic conditions for displacements.

For large strains and geometric nonlinear studies, the periodic boundary conditions read

$$\mathbf{u}_{dst} - \mathbf{u}_{src} = (F_{avg} - I)(\mathbf{X}_{dst} - \mathbf{X}_{src}) \tag{3-207}$$

where F_{avg} is the average deformation gradient.

HOMOGENEOUS BOUNDARY CONDITIONS

For a statically homogeneous microstructure represented by an RVE, the displacement boundary condition is written as

$$\mathbf{u} = \varepsilon_{avg}\mathbf{X}$$

where ε_{avg} is a global average strain.

The displacements on a pair of parallel (and opposite) boundaries is written as

$$\mathbf{u}_{dst} = \varepsilon_{avg}\mathbf{X}_{dst} \text{ and } \mathbf{u}_{src} = \varepsilon_{avg}\mathbf{X}_{src}$$

where $\mathbf{X}_{dst}$ and $\mathbf{u}_{dst}$ are the position and displacement on the destination side, and $\mathbf{X}_{src}$ and $\mathbf{u}_{src}$ are the position and displacement on the source side.

Under large deformations, the displacement boundary condition is written as

$$\mathbf{u} = F_{avg}\mathbf{X}$$

and the displacements on a pair of parallel (and opposite) boundaries is written as

$$\mathbf{u}_{dst} = (F_{avg} - I)\mathbf{X}_{dst} \text{ and } \mathbf{u}_{src} = (F_{avg} - I)\mathbf{X}_{src}$$

For a statistically homogeneous microstructures, RVE, the traction boundary condition is written as

$$\mathbf{T} = \sigma_{avg}\mathbf{X}$$

where σ_{avg} is a global average stress.

The tractions on a pair of parallel (and opposite) boundaries are written as

$$\mathbf{T}_{dst} = \sigma_{avg}\mathbf{X}_{dst} \text{ and } \mathbf{T}_{src} = \sigma_{avg}\mathbf{X}_{src}$$

where $\mathbf{T}_{dst}$ and $\mathbf{T}_{src}$ are the tractions on the destination and source sides, respectively.

There are different options available to enforce the **Homogeneous** or **Periodic** boundary conditions:

- Free Expansion
- Prescribed Average Strain
- Prescribed Average Stress
- Mixed
- Average Deformation Gradient

FREE EXPANSION

In order to determine the homogeneous coefficient of thermal expansion of a heterogeneous material, the material is subjected to unit rise in temperature, while it

is allowed to expand freely. To model this behavior, the global average strain tensor $\bar{\varepsilon}$ in Equation 3-206 is considered as a global degree of freedom that varies freely. The averaged coefficient of thermal expansion α is computed as

$$\alpha = \frac{1}{V}\int_V \frac{\varepsilon}{\Delta T} dV$$

Similarly, to determine the homogeneous coefficient of hygroscopic swelling, the material is subjected to a unit rise in concentration. The averaged coefficient of hygroscopic swelling β is then computed as

$$\beta = \frac{1}{V}\int_V \frac{\varepsilon}{\Delta C} dV$$

If no averaged properties are computed, the free expansion option only computes the average stress and strain tensors by enforcing Equation 3-206 as a constraint.

It is possible to apply either periodic or homogeneous boundary conditions.

PRESCRIBED AVERAGE STRAIN

In order to determine the homogeneous elasticity tensor of a heterogeneous material, the material is subjected to a unit strain in each direction, one-by-one, while keeping the other strain components equal to zero. To model this behavior, the global average strain tensor ε_{avg} in Equation 3-206 is a user input. The average stress is computed from Equation 3-204, and the elasticity tensor is obtained from Equation 3-205. It is possible to apply either periodic or homogeneous boundary conditions.

PRESCRIBED AVERAGE STRESS

In order to determine the homogeneous compliance tensor of a heterogeneous material, the material is subjected to a unit stress in each direction, one-by-one, while keeping the other stress components equal to zero. The average stress is a user input, and the global average strain tensor ε_{avg} in Equation 3-206 is a global degree of freedom, which is defined by a global weak equation

$$\sigma_{avg} : \delta\varepsilon_{avg} \tag{3-208}$$

The homogenized compliance tensor $\bar{C}^{-1}$ is then obtained from the following equation

$$\varepsilon_{avg} = \bar{C}^{-1} : \sigma_{avg}$$

It is possible to apply either periodic or homogeneous boundary conditions.

MIXED

This option studies the response of the unit cell when subjected to a combination of prescribed stress and strain. You can choose which components of the average stress and strain tensors to enter as user inputs; the remaining components are computed from Equation 3-204 and Equation 3-208 by enforcing Equation 3-206 as a constraint. It is only possible to apply periodic boundary conditions.

AVERAGE DEFORMATION GRADIENT

To model this behavior, the average deformation gradient tensor $\boldsymbol{F}_{\text{avg}}$ in Equation 3-207 is a user input. The average first Piola–Kirchhoff stress tensor is computed from

$$P_{\text{avg}} = \frac{1}{V}\int_V P dV \tag{3-209}$$

It is possible to apply either periodic or homogeneous boundary conditions.

Rigid Connector

The rigid connector is a special kinematic constraint, which can be attached to one or several boundaries, edges, or points. The effect is that all connected entities behave as if they were connected by a common rigid body.

There are two different formulations of the rigid connector, *rigid* and *flexible*. In the flexible form, the rigid motion is prescribed only in an average sense, and instead an assumption about linearly distributed traction fields is used.

The only degrees of freedom needed to represent this assembly are the ones needed to represent the movement of a rigid body. In 2D this is simply two in-plane translations, and the rotation around the z-axis.

In 3D the situation is more complex. Six degrees of freedom, usually selected as three translations and three parameters for the rotation, are necessary. For finite rotations, however, any choice of three rotation parameters is singular at some specific set of angles. For this reason, a four-parameter *quaternion* representation is used for the rotations in COMSOL Multiphysics. Thus, each rigid connector in 3D actually has seven degrees of freedom, three for the translation and four for the rotation. The quaternion parameters are called a, b, c, and d. These four parameters are not independent, so an extra equation stating that the following relation is added:

$$a^2 + b^2 + c^2 + d^2 = 1$$

The connection between the quaternion parameters and a rotation matrix R is

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix}$$

Under pure rotation, a vector from the center of rotation ($\mathbf{X}_c$) of the rigid connector to a point $\mathbf{X}$ on the undeformed object is rotated into

$$\mathbf{x} - \mathbf{X}_c = R \cdot (\mathbf{X} - \mathbf{X}_c)$$

where $\mathbf{x}$ is the new position of the point originally at $\mathbf{X}$. The displacement is by definition

$$\mathbf{u} = \mathbf{x} - \mathbf{X} = (R - I) \cdot (\mathbf{X} - \mathbf{X}_c)$$

where I is the unit matrix.

When the center of rotation of the rigid connector also has a translation $\mathbf{u}_c$, then the complete expression for the rigid body displacements is

$$\mathbf{u}_r = (R - I) \cdot (\mathbf{X} - \mathbf{X}_c) + \mathbf{u}_c \tag{3-210}$$

The total rotation of the rigid connector can be also presented as a rotation vector. Its definition is

$$\Theta = \frac{2 \operatorname{atan}\left(\frac{\sqrt{b^2 + c^2 + d^2}}{a} \right)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

The parameter a can be considered as measuring the rotation, while b, c, and d can be interpreted as the orientation of the rotation vector. For small rotations, this relation simplifies to

$$\Theta = 2 \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

The rotation vector is available as the variables `thx_tag`, `thy_tag`, and `thz_tag`. Here *tag* is the tag of the **Rigid Connector** node in the Model Builder tree.

It is possible to apply forces and moments directly to a rigid connector. A force implicitly contributes also to the moment if it is not applied at the center of rotation of the rigid connector. The directions of the forces and moments can either be in space and or follow the rotation of the rigid connector.

WEAK FORMULATION

Instead of enforcing the rigid body constraint on the selected boundaries in the pointwise form, it is possible to use a weak form of the constraint. This is invoked by selecting **Use weak constraints for rigid-flexible connection** in the **Constraint Settings** section for a rigid connector. The weak form is implemented as

$$(\mathbf{u} - \mathbf{u}_r) \cdot \text{test}(\mathbf{F}_{rs}) + \mathbf{F}_{rs} \cdot \text{test}(\mathbf{u} - \mathbf{u}_r) = 0 \tag{3-211}$$

Here $\mathbf{u}$ is the displacement field on the boundaries, $\mathbf{u}_r$ is the rigid body displacement as given by Equation 3-210, and $\mathbf{F}_{rs}$ is the reaction force field (Lagrange multiplier).

The weak formulation cannot be combined with the flexible formulation described below.

FLEXIBLE FORMULATION

The flexible formulation of the rigid connector is based on the weak expression of the constraint, Equation 3-211. In the original weak form, however, the reaction force field can have any distribution, in order to enforce the rigid body motion.

In the flexible formulation, it is assumed that the reaction force field has a linear distribution, given by

$$\mathbf{F}_{rs} = \mathbf{F}_c + \mathbf{F}_d \times (\mathbf{X} - \mathbf{X}_c) \tag{3-212}$$

Here $\mathbf{F}_c$ (unit: N) and $\mathbf{F}_d$ (unit: N/m) are two global vectors, each having three degrees of freedom in 3D. $\mathbf{F}_c$ can be directly interpreted as the reaction force at the center of rotation. $\mathbf{F}_d$ can be considered as a representation of the gradient of the reaction force field.

In the case of geometric nonlinearity, Equation 3-212 is replaced by

$$\mathbf{F}_{rs} = \mathbf{F}_c + \mathbf{F}_d \times (R \cdot (\mathbf{X} - \mathbf{X}_c)) \tag{3-213}$$

where R is the rotation matrix. In 2D, $\mathbf{F}_c$ is a vector with two components, whereas F_d is a scalar.

If the boundaries selected in the rigid connector are not contiguous, then each set of connected boundaries will have its own set of $\mathbf{F}_c$ and $\mathbf{F}_d$ degrees of freedom. The center of rotation, $\mathbf{X}_c$, is then taken as the center of gravity for each individual group of boundaries.

When using the flexible formulation, the reaction force degrees of freedom are named `<physics>.<rigid_tag>.F<c|d><boundary_group><DOF>`, and so on. Examples are `solid.rig1.Fc1x` or `solid.rig2.Fd5z`.

Continuity Condition

The **Continuity** node is a way to connect disconnected parts of an assembly by adding constraints or equations on a shared boundary. Consider a problem with two domains Ω_{src} and Ω_{dst} with a fully or partially shared boundary Γ_{int} such that $\Gamma_{int} = \Gamma_{src} \cap \Gamma_{dst}$, as is schematically shown in Figure 3-40.

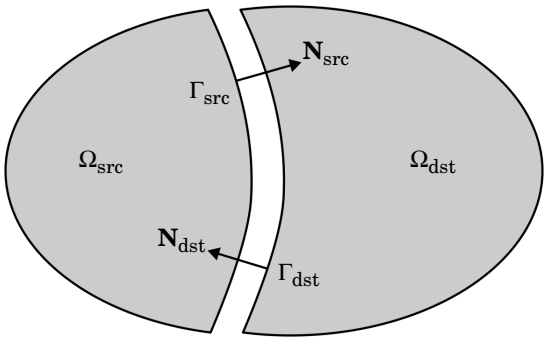


Figure 3-40: Schematic illustration of a continuity interface between two disconnected parts of an assembly.

To enforce continuity between the two domains, the following conditions has to hold on the shared boundary

$$\mathbf{u}_{src} - \mathbf{u}_{dst} = \mathbf{0} \quad \text{on } \Gamma_{int} \tag{3-214}$$

$$\mathbf{P}_{src}\mathbf{N}_{src} + \mathbf{P}_{dst}\mathbf{N}_{dst} = \mathbf{T}_{src} + \mathbf{T}_{dst} = \mathbf{0} \quad \text{on } \Gamma_{int} \tag{3-215}$$

where $\mathbf{P}_i$ is the first Piola–Kirchhoff stress tensor, $\mathbf{T}_i$ is the nominal traction, and $\mathbf{N}_i$ is the normal vector in the material frame for the respective sides of the interface. The

first condition represents the continuity of displacements and the second follows from the Newton's third law. If the **Classical constraints** method in the Continuity node is used, pointwise constraints are set up on Γ_{dst} to enforce the continuity of displacements given by Equation 3-214. This is accomplished with the help of the mapping operator $\text{map}(\boldsymbol{E}, \mathbf{x})$ that is set up by the **Identity pair**, where $\boldsymbol{E}$ is some generic quantity to be mapped and $\mathbf{x}$ is the point to which $\boldsymbol{E}$ is mapped. Using this operator, the constraint equation is written as

$$\text{map}(\mathbf{u}, \mathbf{X}_{\text{dst}}) - \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{\text{dst}}$$

where it is here implied that the mapping is made using material coordinates $\mathbf{X}$, and that the mapping is made from Γ_{src} to Γ_{dst}. The above pointwise constraint implicitly enforces the condition given by Equation 3-215. The above constraint equation can also be implemented as weak constraints by introducing Lagrange multipliers.

Another alternative is to use the **Nitsche** method, which was originally suggested by J. Nitsche in 1971 to weakly impose Dirichlet conditions without having to add Lagrange multipliers. To implement the continuity condition by the Nitsche method, one can, following for example Ref. 160, start from the weak form of a generic boundary condition

$$0 = \ldots + \int_{\Gamma} \mathbf{A}(\mathbf{u}) \cdot \mathbf{B}(\delta\mathbf{u}) dA \tag{3-216}$$

where $\mathbf{A}(\mathbf{u})$ is a flux-like operator and $\mathbf{B}(\mathbf{u})$ is a trace-like operator that can be seen as a conjugate pair, and δ represents the test function. The next step is to reformulate these operators as

$$\mathbf{A}(\mathbf{u}) = \{\mathbf{A}(\mathbf{u}) - \gamma[\mathbf{B}(\mathbf{u}) - \mathbf{c}]\}$$

$$\mathbf{B}(\mathbf{u}) = -\frac{1}{\gamma}[\theta\mathbf{A}(\mathbf{u}) - \gamma\mathbf{B}(\mathbf{u})] + \frac{\theta}{\gamma}\mathbf{A}(\mathbf{u})$$

where $\mathbf{c}$ is a known quantity such as a prescribed displacement, γ is a stabilization factor, and θ is a parameter used to control the symmetry of the formulation. Inserting these reformulations of $\mathbf{A}(\mathbf{u})$ and $\mathbf{B}(\mathbf{u})$ into Equation 3-216, after simplification, leads to the following weak contribution for implementing the Nitsche method for a generic boundary condition

$$0 = \ldots + \theta \int_{\Gamma} \mathbf{B}(\mathbf{u}) \cdot \mathbf{A}(\delta\mathbf{u}) dA + \int_{\Gamma} \mathbf{A}(\mathbf{u}) \cdot \mathbf{B}(\delta\mathbf{u}) dA - \gamma \int_{\Gamma} \mathbf{B}(\mathbf{u}) \cdot \mathbf{B}(\delta\mathbf{u}) dA$$
$$- \int_{\Gamma} \mathbf{c} \cdot [\theta \mathbf{A}(\delta\mathbf{u}) - \gamma \mathbf{B}(\delta\mathbf{u})] dA$$

To implement the continuity condition the unknowns $\mathbf{A}(\mathbf{u})$, $\mathbf{B}(\mathbf{u})$, and $\mathbf{c}$ has to be identified. Considering the continuity of displacements in Equation 3-214, the displacement jump across the interface is defined as

$$\mathbf{u}_{\mathrm{d}} = \begin{cases} \mathbf{u}\text{–map}(\mathbf{u}, \mathbf{X}_{\mathrm{ds}}) & \text{on } \Gamma_{\mathrm{dst}} \\ \mathbf{u}\text{–map}(\mathbf{u}, \mathbf{X}_{\mathrm{src}}) & \text{on } \Gamma_{\mathrm{src}} \end{cases}$$

By also using the average traction defined as

$$\mathbf{T}_{\mathrm{avg}} = \begin{cases} 0{,}5[\mathbf{T} - \mathrm{map}(\mathbf{T}, \mathbf{X}_{\mathrm{dst}})] & \text{on } \Gamma_{\mathrm{dst}} \\ 0{,}5[\mathbf{T} - \mathrm{map}(\mathbf{T}, \mathbf{X}_{\mathrm{src}})] & \text{on } \Gamma_{\mathrm{src}} \end{cases}$$

the action-reaction principle in Equation 3-215 can be weakly imposed as

$$0 = \ldots + \int_{\Gamma} \mathbf{T}_{\mathrm{avg}} \cdot \delta\mathbf{u}_{\mathrm{d}} dA$$

From the above and Equation 3-216, it can identify that

$$\mathbf{A}(\mathbf{u}) = \mathbf{T}_{\mathrm{avg}}$$

$$\mathbf{B}(\mathbf{u}) = \mathbf{u}_{\mathrm{d}}$$

and one can also realize that for continuity, $\mathbf{c} = \mathbf{0}$. The Nitsche formulation of the continuity conditions then follows as

$$0 = \ldots + \theta \int_{\Gamma} \mathbf{u}_{\mathrm{d}} \cdot \delta\mathbf{T}_{\mathrm{avg}} dA + \int_{\Gamma} \mathbf{T}_{\mathrm{avg}} \cdot \delta\mathbf{u}_{\mathrm{d}} dA - \gamma \int_{\Gamma} \mathbf{u} \cdot \delta\mathbf{u}_{\mathrm{d}} dA$$

From the above weak equation several variants of the Nitsche formulation can then be set up. Firstly, it is possible to parameterize the integrals on both sides of the interface,

or alternatively on either Γ_{src} or Γ_{dst} only. The first option is less sensitive to the mesh on either side, but can be more expensive since it involves more mapping operations to evaluate. Secondly, the parameter θ makes it possible to set up formulations with different properties and stability:

- Setting $\theta = 1$ results in a symmetric formulation that maintains the symmetry of the overall system of equations. The main drawback of this setting is that is requires a suitable choice for γ to maintain accuracy and stability.
- Setting $\theta = -1$ results in a skew-symmetric formulation. This formulation is much less sensitive to the choice of γ, but makes the overall system of equations nonsymmetric.
- Setting $\theta = 0$ results in an incomplete formulation. This formulation is less sensitive to the choice of γ than the symmetric one, but not as robust as the skew-symmetric. However, this choice causes the term $\delta\mathbf{T}_{\text{avg}}$ to cancel out. For large deformations and nonlinear materials in particular, this term can be expensive to evaluate, making the nonsymmetric formulation an attractive choice.

Contact Analysis Theory

In this section:

- Contact Search and Kinematics
- Kinematics for Interior Boundaries
- Definition of Contact Tractions
- Normal Contact
- Tangential Contact with Friction
- Unbiased Formulation and Self Contact
- General Contact Equations
- Adhesion
- Decohesion
- Wear

In COMSOL Multiphysics you can model contact between

- A group of boundaries in 1D, 2D, or 3D by adding a Contact Pair and a Contact feature
- A group of geometric objects in 2D or 3D by adding a General Contact Pair and a General Contact feature
- Interior boundaries in 2D or 3D by adding Interior Contact feature

There are three classes of contact methods available: the *penalty* method, the *augmented Lagrangian* method, and the *Nitsche* method. **General Contact** only supports the *penalty* method.

In this section *pair* collectively refers to the **Contact Pair** and the **General Contact Pair**.

- Contact Modeling
- Documentation of the Contact, Friction, Slip Velocity, Adhesion, Decohesion, and Wear features
- Documentation of the General Contact, Contact Model, Damping (General Contact), and Friction (General Contact) features.
- Identity, Contact, and Sector Symmetry Pairs in the *COMSOL Multiphysics Reference Manual*

Contact Search and Kinematics

A prerequisite for setting up the mechanical contact problem is to define the contact search between a source boundary and the destination boundary. The purpose of the search is to detect points on the contacting boundaries that are in contact or may come in contact, and also to map quantities between such points. This part of the contact problem is defined in the pair node that sets up relevant variables and operators to map quantities between the selected source and destination boundaries. In COMSOL Multiphysics, the contact search is made using a ray-tracing strategy as depicted in Figure 3-41. For mechanical contact, variables and equations are formulated on the destination boundary. Hence, the mapping of interest here is from the source

boundary to the destination boundary, however, it is possible to map quantities in the opposite direction as well.

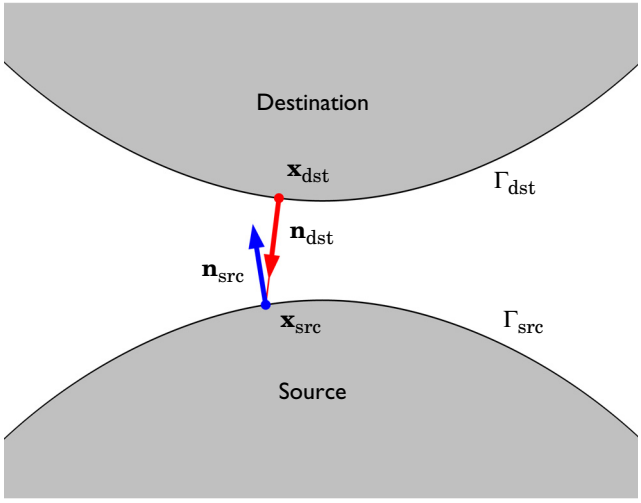


Figure 3-41: Schematic illustration of the contact search and mapping between a source and destination boundary.

For each point $\mathbf{x}_{dst}$ in the spatial frame on the destination boundary Γ_{dst}, the closest point $\mathbf{x}_{src}$ in the spatial frame on the source boundary Γ_{src} in the direction of the destination spatial normal $\mathbf{n}_{dst}$ is sought. Such a mapping can be described through an operator map($\boldsymbol{E}$, $\mathbf{x}$), where $\boldsymbol{E}$ is some generic quantity or expression, and $\mathbf{x}$ is the point to which $\boldsymbol{E}$ is mapped. Using this mapping operator, the coordinates of the mapped point on Γ_{src} is

$$\mathbf{x}_{src} = \text{map}(\mathbf{x}, \mathbf{x}_{dst})$$

where $\mathbf{x}$ is the spatial coordinates that parameterize Γ_{src}. For brevity, the second argument of the mapping operator is omitted in the following and it should be understood that the mapping occurs to a point $\mathbf{x}_{dst}$ on Γ_{dst} if nothing else is specified.

An important quantity for contact mechanics is the gap function $g_{geom}(\mathbf{x})$ defined as the geometric distance between the source and the destination boundary. Note that $g_{geom}(\mathbf{x})$ is not necessarily the closest distance between the two boundaries. Using the mapping operator, the gap distance at $\mathbf{x}_{dst}$ is defined as

$$g_{geom}(\mathbf{x}_{dst}) = -\mathbf{n}_{dst} \cdot [\mathbf{x}_{dst} - \text{map}(\mathbf{x})]$$

With the above definitions, a positive gap infers that there is a geometrical separation between the points, while a negative gap that there is an overclosure. Apart from the mapping operator, the pair also defines a dedicated variable for the gap function with the name `geomgap_dst_<pair_tag>`.

- A **Contact Pair** and a **General Contact Pair** sets up mapping operators that can be used in any expression to map quantities from the source to the destination boundary with the syntax `src2dst_<pair_tag>(expr)`.
- An operator is also defined for mapping quantities from the destination to the source boundary with the syntax `dst2src_<pair_tag>(expr)`. Similarly, a gap variable is defined with the name `geomgap_src_<pair_tag>`.
- For a **General Contact Pair**, the source and destination boundaries are always the same. Hence, the `src2dst_<pair_tag>(expr)` and `dst2src_<pair_tag>(expr)` also give identical results.
- The mapping by default uses the spatial frame normal. If the **Mapping method** in the pair is changed to **Initial configuration**, the mapping is instead performed in the direction of the material frame normal. This means that the direction of the mapping is constant and not affected by how the contacting boundaries deform.

For the mechanical contact problem, it is additionally possible to define a physical offset from the geometric source and destination boundary. This means that the physical gap function $g_n(\mathbf{x})$ and geometrical gap function do not necessarily coincide. The physical gap distance including offsets at $\mathbf{x}_{dst}$ is defined as

$$g_n(\mathbf{x}_{dst}) = g_{geom} - d_{offset,\,d} - \frac{\text{map}(d_{offset,\,s})}{-(\mathbf{n}\cdot\text{map}(\mathbf{n}))} \tag{3-217}$$

where $d_{offset,s}$ and $d_{offset,d}$ is the offset from the source and destination boundary, respectively. Each offset variable can vary along the geometry, and may also vary in time (or by the parameter value when using the continuation solver). The correction of the source side offset is necessary since the normal of $\mathbf{x}_{src}$, $\mathbf{n}_{src} = \text{map}(\mathbf{n})$, is not necessarily pointing toward $\mathbf{x}_{dst}$ as illustrated in Figure 3-41. When the geometric surfaces are in contact, the normals will be exactly opposite such that $\mathbf{n}_{src} = -\mathbf{n}_{dst}$.

In some cases, the discretization of the geometry, especially at curved boundaries, will give small irregularities in the gap function. You can then choose to adjust the physical

gap by removing the geometric gap computed in the initial configuration, g_{init}, smaller than the tolerance Δ_{gap}. In this case, the physical gap function at $\mathbf{x}_{\text{dst}}$ is

$$g_{\text{n}}(\mathbf{x}_{\text{dst}}) = \begin{cases} g_{\text{geom}} - d_{\text{offset, d}} - \dfrac{\text{map}(d_{\text{offset, s}})}{-(\mathbf{n}\cdot\text{map}(\mathbf{n}))} & \text{if } g_{\text{init}} \geq \Delta_{\text{gap}} \\ g_{\text{geom}} - g_{\text{init}} - d_{\text{offset, d}} - \dfrac{\text{map}(d_{\text{offset, s}})}{-(\mathbf{n}\cdot\text{map}(\mathbf{n}))} & \text{else} \end{cases}$$

For problems including friction or adhesion, it is also necessary to know the tangential movement, or slip, between the source and destination boundary. The rate of the relative slip vector $\mathbf{g}_{\text{t}}$ between the two boundaries at $\mathbf{x}_{\text{dst}}$ is defined as

$$\frac{\partial}{\partial t}\mathbf{g}_{\text{t}}(\mathbf{x}_{\text{dst}}) = \text{map}(\mathbf{F}, \mathbf{x}_{\text{dst}})\text{map}\left(\frac{\partial \mathbf{X}}{\partial t}, \mathbf{x}_{\text{dst}}\right) \tag{3-218}$$

where $\mathbf{X}$ is the material coordinates, and $\mathbf{F}$ is the deformation gradient that contains information about the local rotation and stretch. Note that the mapping is made between points in the spatial frame as clarified in the equation. Both quantities are mapped from the source to the destination boundary. Constitutive models for the mechanical problem are typically formulated on an incremental form. COMSOL Multiphysics applies a backward Euler approximation to Equation 3-218 so that the incremental slip at $\mathbf{x}_{\text{dst}}$ is

$$\Delta\mathbf{g}_{\text{t}}(\mathbf{x}_{\text{dst}}) = \text{map}(\mathbf{F})[\text{map}(\mathbf{X}) - \mathbf{X}_{\text{src, old}}]$$

where $\mathbf{X}_{\text{src,old}}$ is the mapped material coordinates at the previous increment.

When a **Contact** node is added to a Shell, Layered Shell, or Membrane interface, the material and spatial frames used in the contact mapping are adjusted to account for the thickness of the structural element by considering the physical position of its top and bottom surface relative to the geometric boundary. This also applies to boundaries that intersect the selection of a **Thin Layer** node in the Solid Mechanics interface. For example, the contact material coordinates $\mathbf{X}^{\text{cnt}}$ and the spatial coordinates $\mathbf{x}^{\text{cnt}}$ in a Contact node added in the Shell interface are defined as

$$\mathbf{X}^{\text{cnt}} = \mathbf{X} + z\mathbf{N}$$

$$\mathbf{x}^{\text{cnt}} = \mathbf{x} + z\mathbf{n}$$

where z is the offset from the geometric boundary to the considered contact boundary. The definition of the geometric gap is then accordingly adjusted and given as

$$g_{\text{geom}}(\mathbf{x}_{\text{dst}}^{\text{cnt}}) = -\mathbf{n}_{\text{dst}} \cdot [\mathbf{x}_{\text{dst}}^{\text{cnt}} - \text{map}(\mathbf{x}^{\text{cnt}}, \mathbf{x}_{\text{dst}}^{\text{cnt}})]$$

where $\mathbf{n}_{\text{dst}}$ is to be interpreted as the normal of the contact boundary. For the Shell interface, the direction of $\mathbf{n}_{\text{dst}}$ coincides with the normal of the geometric boundary, but so is not always the case. This applies to $\mathbf{n}_{\text{src}}$ as well. Note also the that mapping is made between the modified spatial coordinates. Similar corrections are also made for other quantities such as the incremental slip $\Delta\mathbf{g}_{\text{t}}$ where the $\mathbf{X}^{\text{cnt}}$ is used instead of $\mathbf{X}$. The same concept is used for other cases where the contact boundary does not coincide with the geometric boundary, but the definition of $\mathbf{X}^{\text{cnt}}$ and $\mathbf{x}^{\text{cnt}}$ may differ.

Kinematics for Interior Boundaries

When a contact condition is defined on interior boundaries by adding an **Interior Contact** node, a slit condition is defined for the displacement field. All variables and equations are defined for the midsurface of the boundary, and there is thus no distinction made between source and destination boundaries.

The gap function $g_{\text{geom}}(\mathbf{X})$ is here a function of the material coordinates $\mathbf{X}$, and is directly defined by the displacements as

$$g_{\text{geom}}(\mathbf{X}) = \mathbf{n}(\mathbf{X}) \cdot (\mathbf{u}_{\text{u}} - \mathbf{u}_{\text{d}})$$

Here $\mathbf{u}_{\text{u}}$ and $\mathbf{u}_{\text{d}}$ are the displacements on the "upside" and "downside" of the boundary, respectively. The spatial normal $\mathbf{n}$ is always evaluated at the midsurface of the slit boundary. The physical gap distance including offsets is then

$$g_{\text{n}}(\mathbf{X}) = g_{\text{geom}} - d_{\text{offset}}$$

A negative offset, d_{offset}, thus implies that there is an initial gap, while a positive offset implies that there is an initial overclosure.

Similarly, the tangential deformation $\mathbf{g}_{\text{t}}$ is given by

$$\mathbf{g}_{\text{t}}(\mathbf{X}) = \mathbf{t}(\mathbf{X}) \cdot (\mathbf{u}_{\text{u}} - \mathbf{u}_{\text{d}})$$

where $\mathbf{t}$ is the spatial tangent to the interior boundary. Friction is based on an incremental formulation, and the incremental slip is given by

$$\Delta\mathbf{g}_{\text{t}} = \mathbf{g}_{\text{t}} - \mathbf{g}_{\text{t,old}}$$

where $\mathbf{g}_{\text{t,old}}$ is the tangential deformation at the previous converged increment.

Since the gap and the slip are computed from the displacements only, the contact formulation on interior boundaries is only valid for small sliding. Hence, contact on interior boundaries does not enforce geometric nonlinearity.

Definition of Contact Tractions

Since contact enforces geometric nonlinearity, contact tractions (for example the contact pressure and friction forces) are represented as nominal tractions. They are thus to be viewed as a density with components in the current configuration (that is, on the spatial frame) but related to the undeformed area of the corresponding surface element.

Normal Contact

The contact constraint in the normal direction to the contacting boundaries can be represented by the nonpenetration condition. On the destination boundary Γ_{dst}, the normal contact condition can in a weak sense be written as (the contribution of normal contact to the virtual work)

$$0 = \int_{\Gamma_{dst}} T_n \delta g_n dA \tag{3-219}$$

where T_n is the contact pressure, assumed positive in compression. To complete the contact condition, Equation 3-219 is subjected to the Kuhn–Tucker conditions

$$\begin{aligned} g_n &\geq 0 \\ T_n &\geq 0 \\ T_n g_n &= 0 \end{aligned} \tag{3-220}$$

where the first inequality represents the nonpenetration condition. Equation 3-219 and Equation 3-220 together represents a unilateral constraint that require special techniques to implement. Several different techniques for doing so are available in COMSOL Multiphysics, which can be divided into three main classes: the *penalty method*, the *augmented Lagrangian method*, and the *Nitsche method*.

An alternative way of expressing the contact condition is to view the contact pressure as a force acting on both source and destination described as a distributed load. Making use of Newton's second law, an alternative to Equation 3-219 can then be formulated as

$$0 = -\int_{\Gamma_{dst}} T_n \mathbf{n}_{dst} \cdot [\delta \mathbf{u} - \text{map}(\delta \mathbf{u})] dA \tag{3-221}$$

In principle, Equation 3-219 and Equation 3-221 describes the same contribution to the virtual work. However, when the system of equations are linearized they will produce different results. The main difference originates from the variation δg_n with respect to $\mathbf{u}$ that, for the ray-tracing mapping strategy used by COMSOL Multiphysics, includes several terms that can be difficult and expensive to evaluate. These are not present if Equation 3-221 is used, which can increase the robustness of the solution. What formulation that is used can be controlled by the **Jacobian contribution** setting in the **Contact**, **General Contact** or **Contact Model** nodes. When set to **Symmetric** or **Legacy**, Equation 3-219 is used. For **Symmetric**, some terms of δg_n are omitted to improve the robustness, whereas **Legacy** uses the full variation of δg_n. When the **Nonsymmetric** option is selected, Equation 3-221 is used.

The Penalty Method

For the penalty method, the contact pressure in Equation 3-219 is represented using a penalization of the physical gap so that

$$T_n = p_n T(g_n, \ldots) \tag{3-222}$$

where p_n is the penalty factor, $T(g_n, \ldots)$ is a dimensionless penalty function of the gap and additional parameters. This leads to an approximative solution of the nonpenetration condition. The penalty method as defined by Equation 3-222 is conceptually based on the insertion of a nonlinear spring between the contacting points. The penalty factor can be interpreted as the stiffness of that spring. A high value of the stiffness enforces the contact conditions more strictly, but if it is too high the problem becomes ill conditioned and unstable.

The standard choice of penalty function is to use ramp function. Equation 3-222 then specializes to

$$T_n = \begin{cases} -p_n g_n + T_0 & \text{if } g_n < \dfrac{T_0}{p_n} \\ 0 & \text{otherwise} \end{cases}$$

The pressure at zero gap, T_0, can be used to reduce the overclosure between the contacting boundaries if an estimate to the contact pressure is known. In the default case, when $T_0 = 0$, there must always be some overclosure in order for a contact pressure to develop.

An alternative is to define the penalty function using a smooth penalty function. Equation 3-222 then specializes to

$$T_{\mathrm{n}} = \begin{cases} p_{\mathrm{n}} T(\, g_{\mathrm{n}}, \Delta_{\mathrm{start}}, \Delta_{\mathrm{size}}) & \text{if } g_{\mathrm{n}} < \Delta_{\mathrm{start}} \\ 0 & \text{otherwise} \end{cases}$$

where $T(g_n, \Delta_{\mathrm{start}}, \Delta_{\mathrm{size}})$ defines a smooth ramp function with a transition zone defined by two parameters. Here, the built-in Ramp function is used with a smooth transition zone at the start with 2 continuous derivatives. The start of the transition zone is given by Δ_{start}, and its size is given by Δ_{size}. A smooth penalty function is often useful to stabilize the initiation of dynamic contact events such as impact.

In addition to Equation 3-222, the **Contact** feature provides an alternative penalization based on a viscous formulation intended for transient simulations, referred to as the **Penalty, dynamic** method. This formulation is based on the additional persistency condition

$$T_{\mathrm{n}} \frac{\partial g_{\mathrm{n}}}{\partial t} = 0$$

In this case, the nonpenetration and persistency conditions are enforced by penalizing the rate of the physical gap distance such that the contact pressure is defined by

$$T_{\mathrm{n}} = \begin{cases} p_{\mathrm{n}} \langle -\frac{\partial g_{\mathrm{n}}}{\partial t} \rangle & \text{if } g_{\mathrm{n,\,old}} \le 0 \\ 0 & \text{otherwise} \end{cases} \tag{3-223}$$

where $g_{\mathrm{n,old}}$ is the physical gap distance at the previous increment, and <> is the positive parts operator. With the definition in Equation 3-223, T_{n} is only nonzero if the physical gap is negative during the entire increment, which is important for energy conservation during the initial contact event. However, there will always be some initial penetration before the gap rate is penalized. Hence the observed overclosure will be dependent on the size of the increment when two points come into contact. Note that this definition of the contact pressure implies that pressure contact is dissipative.

It is possible to use the definitions of T_{n} in Equation 3-222 and Equation 3-223 simultaneously. In such a case, they are additive and conceptually represent a parallel coupling of a spring and a dashpot.

For **General Contact**, the contribution to T_n by Equation 3-223 is made by a **Damping** subnode to the contact model.

The weak contribution added for normal contact when using the penalty method is

$$0 = \ldots + \int_{\Gamma_{dst}} T_n \delta g_n dA$$

or

$$0 = \ldots - \int_{\Gamma_{dst}} T_n \mathbf{n}_{dst} \cdot [\delta \mathbf{u} - \mathrm{map}(\delta \mathbf{u})] dA$$

THE AUGMENTED LAGRANGIAN METHOD

The augmented Lagrangian method can be seen as a compromise between using the penalty method and enforcing the nonpenetration condition using weak constraints and Lagrange multipliers. The augmented Lagrangian formulation for normal contact in COMSOL Multiphysics is based on an integral formulation that in a weak sense enforces the nonpenetration condition exactly over the contacting boundaries. It is based on the following augmentation of Equation 3-219

$$0 = \ldots + \int_{\Gamma_{dst}} T_{np} \delta g_n dA - \frac{1}{p_n} \int_{\Gamma_{dst}} (T_{np} - T_n) \delta T_n dA \tag{3-224}$$

where T_{np} is the penalized (or augmented) contact pressure, and T_n is a Lagrange multiplier that can be identified as a contact pressure. Alternatively, Equation 3-221 can be augmented, giving the following result

$$\begin{aligned} 0 = \ldots &- \int_{\Gamma_{dst}} T_{np} \mathbf{n}_{dst} \cdot [\delta \mathbf{u} - \mathrm{map}(\delta \mathbf{u})] dA \\ &- \frac{1}{p_n} \int_{\Gamma_{dst}} (T_{np} - T_n) \delta T_n dA \end{aligned} \tag{3-225}$$

The Lagrange multiplier T_n is defined as a dependent variable of the contact problem and is typically discretized using Lagrange shape functions. There are two methods for solving the above coupled system of equations, and this choice affects the definition of T_{np}.

The first solution method is referred to as a **Segregated** method and corresponds to the so-called Uzawa algorithm. Here, the system of equations is decoupled, and the displacement field $\mathbf{u}$ and the Lagrange multiplier T_n are solved in a segregated way. This leads to two levels of iterations in the solver, where T_n is held constant when solving for $\mathbf{u}$, and vice versa when updating T_n. For this solution method, the penalized contact pressure T_np is defined with a smooth transition to stabilize the problem when transitioning between contact states. The definition for T_np for outer iteration $j+1$ is defined by

$$T_{\mathrm{np},j+1} = \begin{cases} T_{\mathrm{n},j} - p_\mathrm{n} g_\mathrm{n} & \text{if } g_\mathrm{n} \le 0 \\ T_{\mathrm{n},j}\, e^{-\frac{p_\mathrm{n} g_{\mathrm{n_g}}}{T_\mathrm{n}}} & \text{else} \end{cases}$$

The segregated method can then for increment $n+1$ schematically be described as

1. Initialize increment $n+1$ by setting the outer iteration counter j=0, and the initial values for dependent variables $\mathbf{u}_j^{n+1} = \mathbf{u}^n$ and $T_{\mathrm{n},j}^{n+1} = T_\mathrm{n}^n$
2. Compute the displacement $\mathbf{u}_{j+1}$ by solving, for example,

$$0 = \ldots + \int_{\Gamma_\mathrm{dst}} T_{\mathrm{np},j+1} \delta g_{\mathrm{n},j+1} dA$$

3. Compute the Lagrange multiplier $T_{\mathrm{n},j+1}$ by solving

$$0 = \int_{\Gamma_\mathrm{dst}} (T_{\mathrm{np},j+1} - T_{\mathrm{n},j+1}) \delta T_\mathrm{n} dA$$

4. If convergence, continue to the next increment, else return to step 2

The appropriate solver sequence is automatically set up by COMSOL Multiphysics for the **Segregated** solution method when adding a new solver, or resetting to the default solver.

The alternative is to use a **Fully Coupled** solution method, which, as the name implies, amounts to solving the coupled system in Equation 3-224 or Equation 3-225 for the two dependent variables $\mathbf{u}$ and T_n (and any other variables included in the multiphysics model). When using this method, the penalized contact pressure is defined by

$$T_{\mathrm{np},j+1} = \begin{cases} T_{\mathrm{n},j} - p_{\mathrm{n}} g_{\mathrm{n}} & \text{if } g_{\mathrm{n}} \le 0 \\ 0 & \text{else} \end{cases}$$

The **Fully Coupled** method can be convenient as it enforces the nonpenetration condition exactly over the contacting boundaries, while it puts no restriction on how to set up the solver sequence. This flexibility can be especially advantageous when working with multiphysics problems, compared to the segregated approach. However, one should be aware that the method comes with some of the drawbacks of a pure Lagrange multiplier approach, albeit to a lesser extent. One drawback is that the solution typically is a saddle point, and convergence can be sensitive to how Newton iterations are made when solving the nonlinear problem.

In the limit, the **Segregated** and the **Fully Coupled** solution methods converge to the same solution, but they can have different convergence properties.

In addition to the standard augmented Lagrangian method outlined above, an **Augmented Lagrangian, dynamic** method is available for transient simulations. As discussed for the penalty method, this formulation is based on the addition of the persistency condition. A segregated solution method is recommended for this method, and the main difference compared to the standard method lies in the definition of the penalized contact pressure. Both the nonpenetration and persistency conditions are enforced by penalizing the rate of the physical gap distance so that

$$T_{\mathrm{np},j+1} = \begin{cases} \langle T_{\mathrm{n},j} - p_{\mathrm{n}} \dfrac{\partial g_{\mathrm{n}}}{\partial t} \rangle & \text{if } g_{\mathrm{n,\,old}} \le 0 \\ 0 & \text{else} \end{cases} \tag{3-226}$$

where $g_{\mathrm{n,old}}$ is the physical gap distance at the previous increment, and <> is the positive parts operator. The same segregated solution strategy as described above is applied also when the dynamic method is used.

THE NITSCHE'S METHOD

The Nitsche's method for contact is an extension of the general method suggested by J. Nitsche in 1971 to impose Dirichlet conditions in a weak sense without having to add Lagrange multipliers. A general derivation of the method for such applications is presented in Continuity Condition. Here the method is applied to large deformation contact using Equation 3-221 as a starting point, and following similar steps as in Ref. 161 to derive the final equations. The first step is to reformulate the contact pressure from Equation 3-222 as

$$T_{\mathrm{n}} = \langle -(\mathbf{P}\cdot\mathbf{N})\mathbf{n} - p_{\mathrm{n}}\, g_{\mathrm{n}}\rangle = \langle -\mathbf{T}_{\mathrm{a}}\cdot\mathbf{n} - p_{\mathrm{n}}\, g_{\mathrm{n}}\rangle$$

where $\mathbf{P}$ is the first Piola–Kirchhoff stress, $\mathbf{N}$ is the material normal, $\mathbf{n}$ is the spatial normal, and $\mathbf{T}_{\mathrm{a}}$ is the nominal traction vector. The positive parts operator is used above to indicate that T_{n} is strictly positive. The term from Equation 3-221 in brackets including $\delta\mathbf{u}$ is also reformulated as

$$\delta\mathbf{u} - \mathrm{map}(\delta\mathbf{u}) = -\frac{1}{p_{\mathrm{n}}}[p_{\mathrm{n}}\delta\mathbf{u} - p_{\mathrm{n}}\mathrm{map}(\delta\mathbf{u}) - \theta\delta\mathbf{T}_{\mathrm{a}}] + \frac{\theta}{p_{\mathrm{n}}}\delta\mathbf{T}_{\mathrm{a}}$$

where θ is a parameter used to control the symmetry of the formulation. Using the above definitions, Equation 3-221 can, after some manipulation and simplification, be rewritten to obtain the weak contribution added by the Nitsche method for normal contact

$$\begin{aligned} 0 = \ldots &- \frac{1}{p_{\mathrm{n}}}\int_{\Gamma_{\mathrm{dst}}} T_{\mathrm{n}}\mathbf{n}_{\mathrm{dst}}\cdot[\delta\mathbf{u} - \mathrm{map}(\delta\mathbf{u})]\,dA \\ &+ \frac{\theta}{p_{\mathrm{n}}}\int_{\Gamma_{\mathrm{dst}}} (\mathbf{T}_{\mathrm{a}} + T_{\mathrm{n}}\mathbf{n}_{\mathrm{dst}})\cdot(\delta\mathbf{P}\cdot\mathbf{N})\,dA \end{aligned} \tag{3-227}$$

From the above weak equation several variants of Nitsche's formulation for contact can be set up using different values for the parameter θ:

- Setting $\theta = 1$ results in a symmetric formulation. The formulation requires a suitable choice for p_{n} to maintain accuracy and stability. However, it is the most consistent formulation.
- Setting $\theta = -1$ results in a skew-symmetric formulation. This formulation is less sensitive to the choice of p_{n}.
- Setting $\theta = 0$ results in an incomplete formulation. This is less sensitive to the choice of p_{n} than the symmetric formulation, however it is not as robust as the skew-symmetric formulation. This incomplete formulation cancel out the term proportional to $\delta\mathbf{P}$, and since this term can be expensive to evaluate, the incomplete formulation is an attractive choice for contact with nonlinear materials.

Note that since the derivation Equation 3-227 started from Equation 3-221, all of the above choices of θ will cause a nonsymmetric stiffness matrix when setting of the global system of equations. Also, the penalty factor p_{n} for the Nitsche's method can be considered as a stabilization factor, not directly as a spring stiffness as in the penalty method.

DISSIPATION

The **Penalty, dynamic** and **Augmented Lagrange, dynamic** methods in the **Contact** feature are based on a viscous type formulation, and dissipate energy when the contact is active. The accumulated dissipated energy W_{cntv} can be computed by solving a distributed ODE on the destination boundary:

$$\frac{\partial}{\partial t} W_{\text{cntv}} = T_{\text{n}} \frac{\partial g_{\text{n}}}{\partial t}$$

Similarly the **Damping** node in general contact dissipates energy. For general contact, the dissipated energy W_{cnt} is defined by internal state variables.

Tangential Contact with Friction

The tangential behavior of a contact interface, due to friction, can be divided into two states, the stick state and the slip state. The stick state is simply defined by the constraint that there can be no tangential movement between the two contacting boundaries. When the contact enters the slip state, the tangential condition enforces the direction of the relative tangential motion between points $\mathbf{x}_{\text{dst}}$ and $\mathbf{x}_{\text{src}}$ on the respective side of the interface. On the destination boundary Γ_{dst}, the tangential contact condition can be written as a weak equation (the contribution of tangential contact to the virtual work)

$$0 = \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{t}} \cdot \delta \mathbf{g}_{\text{t}} dA \tag{3-228}$$

where $\mathbf{T}_{\text{t}}$ is the friction force. The stick and slip conditions are then enforced by the constitutive model used to define $\mathbf{T}_{\text{t}}$.

Using the same concepts as for the normal contact, and alternative to Equation 3-228 is to use the following weak equation

$$0 = \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{t}} \cdot [\delta \mathbf{u} - \text{map}(\delta \mathbf{u})] dA \tag{3-229}$$

CONSTITUTIVE MODEL FOR FRICTION

The constitutive model for friction, including stick and slip, is based on the Coulomb law. In a general framework similar to plasticity theory, the model can be expressed as

$$F_{\text{slip}} \equiv \|\mathbf{T}_{\text{t}}\| - T_{\text{t, crit}} \leq 0$$
$$\frac{\partial \mathbf{g}_{\text{t}}}{\partial t} = \lambda \frac{\mathbf{T}_{\text{t}}}{\|\mathbf{T}_{\text{t}}\|} \tag{3-230}$$
$$\lambda \geq 0$$
$$\lambda F_{\text{slip}} = 0$$

where F_{slip} is the slip function (analogue to the yield function in plasticity theory), and λ is a friction multiplier. The critical friction force $T_{\text{t,crit}}$ determines when slip occurs, and is defined as

$$T_{\text{t, crit}} = \min(T_{\text{cohe}} + \mu T_{\text{n}}, T_{\text{t, max}})$$

where T_{cohe} is the cohesion, and $T_{\text{t,max}}$ defines the maximum friction force that is admissible. The friction coefficient μ can either be defined as an arbitrary constant expression or through an exponential decay model in time-dependent studies. The latter model is defined as

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}}) \exp(-\alpha_{\text{dcf}} \|\mathbf{v}_{\text{s}}\|)$$

where μ_{stat} is the static friction coefficient, μ_{dyn} is the dynamic friction coefficient, $\mathbf{v}_s$ is the slip velocity, and α_{dcf} is a decay coefficient.

It is possible to also define the critical friction force through an arbitrary expression. In principle, $T_{\text{t,crit}}$, or any variable described above, may be defined as a function of any other variable or field. However, the implementation described in the following is only valid as long as it does not dependent on $\mathbf{T}_{\text{t}}$. In other words, the implementation of the friction constitutive model does not allow for hardening behavior of the friction force.

THE PENALTY METHOD

In COMSOL Multiphysics, implementation of Equation 3-230 is made using a backward Euler integration. Together with a penalty regularization, this results in the following set of algebraic equations to solve at increment $n + 1$:

$$F_{\text{slip}, n+1} \equiv \|\mathbf{T}_{\text{t}, i+1}\| - T_{\text{t, crit}, n+1} \leq 0$$
$$\mathbf{T}_{\text{t}, n+1} = \mathbf{T}_{\text{t}, n} + p_{\text{t}} \left(\Delta \mathbf{g}_{\text{t}} - \lambda \frac{\mathbf{T}_{\text{t}, n+1}}{\|\mathbf{T}_{\text{t}, n+1}\|} \right)$$
$$\lambda \geq 0$$
$$\lambda F_{\text{slip}, n+1} = 0$$

where $\mathbf{T}_{t,n}$ is the friction force at the previous increment. The set of algebraic equations is solved for unknowns $\mathbf{T}_{t,n+1}$ and λ by introducing a trial state and a return mapping. In doing so, it is assumed that $T_{t,\text{crit},n+1}$ is independent of $\mathbf{T}_{t,n+1}$, which leads to an explicit set of equations for finding $\mathbf{T}_{t,n+1}$. The trial state is first computed as

$$\mathbf{T}_{t,n+1}^{\text{trial}} = \mathbf{T}_{t,n} - p_t \Delta \mathbf{g}_t$$

and the return mapping gives the current value of the friction force

$$\mathbf{T}_{t,n+1} = \begin{cases} \mathbf{T}_{t,n+1}^{\text{trial}} & \text{if } F_{\text{slip},n+1} \le 0 \\ T_{t,\text{crit},n+1} \dfrac{\mathbf{T}_{t,n+1}^{\text{trial}}}{\left\| \mathbf{T}_{t,n+1}^{\text{trial}} \right\|} & \text{if } F_{\text{slip},n+1} > 0 \end{cases} \tag{3-231}$$

The weak contribution added for tangential contact when using the penalty method follows from Equation 3-228, and is given by

$$0 = \ldots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{t,n+1} \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

or from Equation 3-229 as

$$0 = \ldots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{t,n+1} \cdot [\delta \mathbf{u} - \text{map}(\delta \mathbf{u})] dA$$

THE AUGMENTED LAGRANGIAN METHOD

The augmented Lagrangian formulation for tangential contact in COMSOL Multiphysics is based on an integral formulation that in a weak sense enforces the stick and slip conditions exactly over the contacting boundaries. It is based on the following augmentation of Equation 3-228

$$0 = \ldots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{tp} \cdot \delta \mathbf{g}_t dA + \frac{1}{p_t} \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{tp} - \mathbf{T}_t) \cdot \delta \mathbf{T}_t dA \tag{3-232}$$

where $\mathbf{T}_{tp}$ is the penalized (or augmented) friction, and $\mathbf{T}_t$ is a Lagrange multiplier that can be identified as a traction. Alternatively, Equation 3-229 can be augmented, giving the following result

$$0 = \ldots + \int_{\Gamma_{dst}} \mathbf{T}_{tp} \cdot [\delta\mathbf{u} - \text{map}(\delta\mathbf{u})] dA + \frac{1}{p_t} \int_{\Gamma_{dst}} (\mathbf{T}_{tp} - \mathbf{T}_t) \cdot \delta\mathbf{T}_t dA \tag{3-233}$$

The Lagrange multiplier $\mathbf{T}_t$ is defined as the dependent variable of the contact problem and is typically discretized using Lagrange shape functions. There are two methods for solving the above coupled system of equations, **Segregated** and **Fully coupled**, as selected in the parent **Contact** node.

As for the penalty method, Equation 3-230 is implemented using backward Euler integration of the evolution equations, and a trial state and return mapping to solve the nonlinear algebraic equations.

When the **Segregated** solution method is selected in the parent **Contact** node, the trial state for outer iteration $j+1$ is computed as

$$\mathbf{T}_{t,j+1}^{trial} = \mathbf{T}_{t,j} - p_t \Delta\mathbf{g}_t$$

and the return mapping gives the current value of the penalized friction force

$$\mathbf{T}_{tp,j+1} = \begin{cases} \mathbf{T}_{t,j+1}^{trial} & \text{if } F_{slip,j+1} \le 0 \\ T_{t,crit,j+1} \dfrac{\mathbf{T}_{t,i+1}^{trial}}{\left\| \mathbf{T}_{t,j+1}^{trial} \right\|} & \text{if } F_{slip,j+1} > 0 \end{cases}$$

This scheme then fits in to the algorithmic format described for normal contact where the displacement field $\mathbf{u}$ and the Lagrange multipliers associated with contact are solved in a segregated way. Including friction, this algorithm is schematically described as

1. Initialize increment $n+1$ by setting the outer iteration counter $j = 0$, and the initial values for dependent variables $\mathbf{u}_j^{n+1} = \mathbf{u}^n$, $T_{n,j}^{n+1} = T_n^n$, and $\mathbf{T}_{t,j}^{n+1} = \mathbf{T}_t^n$
2. Compute the displacement $\mathbf{u}_{j+1}$ by solving for example

$$0 = \ldots + \int_{\Gamma_{dst}} T_{np,j+1} \delta g_n dA + \int_{\Gamma_{dst}} \mathbf{T}_{tp,j+1} \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

3. Compute the Lagrange multipliers $T_{n,j+1}$ and $\mathbf{T}_{t,j+1}$ by solving

$$0 = \int_{\Gamma_{\text{dst}}} (T_{\text{np},j+1} - T_{\text{n},j+1})\delta T_{\text{n}} dA + \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{\text{tp},j+1} - \mathbf{T}_{\text{t},j+1})\delta T_{\text{n}} dA$$

4 If convergence, continue to the next increment, else return to step 2

The segregated algorithm is also used when friction is used together with the **Augmented Lagrangian, dynamic** formulation.

When the **Fully coupled** solution method is selected in the parent **Contact** node, the trial friction force for iteration $i+1$ is computed as

$$\mathbf{T}_{\text{t},i+1}^{\text{trial}} = \mathbf{T}_{\text{t},i} - p_{\text{t}}\Delta\mathbf{g}_{\text{t}}$$

The important difference here lies in that for the coupled method, the trial state is computed as the change from the previous nonlinear iteration every time the coupled problem is solved. Knowing the trial state, the return mapping is performed similar to the other methods.

The weak contribution added for tangential contact when using the **Fully coupled** solution method follows from Equation 3-232, and is

$$0 = \ldots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{t}} \cdot \text{map}(\mathbf{F})\delta[\text{map}(\mathbf{X})] dA + \frac{1}{p_{\text{t}}} \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{\text{tp}} - \mathbf{T}_{\text{t}})\delta\mathbf{T}_{\text{t}} dA$$

or similarly from the alternative augmentation in Equation 3-233.

THE NITSCHE'S METHOD

The derivation of Nitsche's formulation for tangential contact with friction follows similar steps as for the normal contact in The Nitsche's Method. Replacing the contact pressure vector $-T_{\text{n}}\mathbf{n}$ in Equation 3-227 with the friction force $\mathbf{T}_{\text{t}}$, the weak contribution becomes

$$0 = \ldots + \frac{1}{p_{\text{n}}} \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{t}} \cdot [\delta\mathbf{u} - \text{map}(\delta\mathbf{u})] dA + \frac{\theta}{p_{\text{n}}} \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{\text{a}} - \mathbf{T}_{\text{t}}) \cdot (\delta\mathbf{P} \cdot \mathbf{N}) dA$$

The definition of $\mathbf{T}_{\text{t}}$ is done in a similar way as for the penalty method following Equation 3-231. However, for Nitsche's formulation of frictional contact, the trial friction force is given as

$$\mathbf{T}_{\mathrm{t},\,n+1}^{\mathrm{trial}} = \mathbf{P}\cdot\mathbf{N} - (\mathbf{P}\cdot\mathbf{N})\mathbf{n} - p_{\mathrm{n}}\Delta\mathbf{g}_{\mathrm{t}} = \mathbf{T}_{\mathrm{at}} - p_{\mathrm{n}}\Delta\mathbf{g}_{\mathrm{t}}$$

where $\mathbf{T}_{\mathrm{at}}$ is the tangential part of the nominal traction vector. Note that the same penalty factor, p_{n}, is always used for the normal and tangential parts of the contact problem when using the Nitsche method.

PRESCRIBED SLIP VELOCITY

In the Slip Velocity subnode to **Contact**, it is assumed the tangential contact is in a slip state when in contact, and that the relative motion between the source and destination boundaries is known beforehand. That is

$$\frac{\partial \mathbf{g}_{\mathrm{t}}}{\partial t} = \mathbf{v}_{\mathrm{slip}}$$

where $\mathbf{v}_{\mathrm{slip}}$ is the prescribed velocity on the destination boundary relative to the source boundary. Given that the rate of the slip is known, the constitutive model for friction in Equation 3-230 is simplified, and it follows that

$$\mathbf{T}_{\mathrm{t}} = T_{\mathrm{t,\,crit}}\frac{-\mathbf{v}_{\mathrm{slip}}}{\|\mathbf{v}_{\mathrm{slip}}\|}$$

since the slip velocity and friction force are collinear. The weak contribution added for tangential contact by the slip velocity follows from Equation 3-228 and is given by

$$0 = \ldots + \int_{\Gamma_{\mathrm{dst}}} \mathbf{T}_{\mathrm{t}}\cdot\mathrm{map}(\mathbf{F})\delta[\mathrm{map}(\mathbf{X})]\,dA$$

or from Equation 3-229 as

$$0 = \ldots + \int_{\Gamma_{\mathrm{dst}}} \mathbf{T}_{\mathrm{t}}\cdot[\delta\mathbf{u} - \mathrm{map}(\delta\mathbf{u})]\,dA$$

DISSIPATION

Friction is a dissipative process and the dissipated energy W_{fric} can be computed by solving a distributed ODE on the destination boundary:

$$\frac{\partial}{\partial t}W_{\mathrm{fric}} = -\mathbf{T}_{\mathrm{t}}\cdot\frac{\partial \mathbf{g}_{\mathrm{t}}}{\partial t}$$

For general contact, the dissipated energy W_{fric} is defined by internal state variables.

Unbiased Formulation and Self Contact

The standard formulation for contact in COMSOL Multiphysics as described in Normal Contact and Tangential Contact with Friction can be viewed as a biased (or unsymmetric) formulation in the sense the contact forces are added on one side of a contact pair only — the destination boundary. Often this is a preferred formulation because it involves the fewest possible evaluation points where equations and contact mappings have to be evaluated, making it attractive from a performance point of view. However, it puts requirements on, for example, the relative mesh density between source and destination to be accurate. In some situations it might also be difficult to *a priori* split a set of boundaries in a source and destination as, for example, when objects are self-intersecting. This situation is typically referred to as self contact.

For such cases, it is also possible to set up an unbiased (or symmetric) formulation for contact by using the same selection as source and destination. That is, $\Gamma_{\text{src}} = \Gamma_{\text{dst}}$ in the schematic problem description in Figure 3-41 and one can instead refer to a single contact boundary Γ_{cnt}. Since contact forces will then be added on both sides of a pair of mapped points, the added equations needs to be modified in order to be consistent. To accomplish this, Equation 3-219 is instead written as

$$0 = \frac{1}{2}\int_{\Gamma_{\text{cnt}}} T_{\text{n}}\delta g_{\text{n}} dA$$

when it is detected that $\Gamma_{\text{src}} = \Gamma_{\text{dst}}$. Similarly Equation 3-221 is formulated as

$$0 = -\frac{1}{2}\int_{\Gamma_{\text{cnt}}} T_{\text{n}}\mathbf{n}_{\text{dst}} \cdot [\delta\mathbf{u} - \text{map}(\delta\mathbf{u})] dA$$

The specialized versions of these equations for the penalty method, the augmented Lagrangian method, and the Nitsche method as well as those for tangential contact then follows similarly.

General Contact Equations

A key aspect of the problem set up for general contact is that the **General Contact Pair** always defines the source and destination as the same selection. Hence a more formal definition of source and destination is needed in the following:

- The *source* is a boundary that is included in the contact search.

- The *destination* is a boundary from which the contact search is made and where equations are added.
- A boundary can be both source and destination at the same time.

With this definition, the destination is identified as the set of boundaries given by the intersection of the boundaries included in the **General Contact Pair** and a **General Contact** node. That is, there may be source boundaries that are not part of the destination.

To obtain a consistent formulation, general contact automatically detects if a source of a contact mapping is also part of the destination boundaries and either applies the biased formulation as given by Normal Contact and Tangential Contact with Friction or unbiased formulation as described by Unbiased Formulation and Self Contact. This means that self-contact is automatically accounted for by general contact.

General contact always uses the penalty method to implement the contact conditions. Hence, the weak equation can be summarized as

$$0 = \int_{\Gamma_{cnt}} \Lambda T_n \delta g_n dA + \int_{\Gamma_{cnt}} \Lambda \mathbf{T}_t \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

or

$$0 = -\int_{\Gamma_{cnt}} \Lambda \mathbf{T} \cdot [\delta \mathbf{u} - \text{map}(\delta \mathbf{u})] dA$$

where $\Lambda = 0.5$ if the source of the mapping is part of the destination boundaries, otherwise $\Lambda = 1.0$. The definition of the contact tractions follows Normal Contact and Tangential Contact with Friction.

Adhesion

You can model a situation where two boundaries stick together once they get into contact by adding an Adhesion subnode to **Contact**. Adhesion can only be modeled when the penalty contact method is used. The adhesion formulation can be viewed as if a thin elastic layer is placed between the source and destination boundaries when adhesion is activated.

The adhesion starts acting when the *adhesion criterion* is met in the previous time or parameter step. An internal degree of freedom located at Gauss points is used as an indicator of whether the adhesion criterion has been met or not.

Enter a Boolean expression in the **Reset criterion** field to deactivate the adhesive force, such that adhesion is removed when it evaluates to true.

Using the effective gap distance g_n and the slip $\Delta \mathbf{g}_t$, the adhesion formulation defines an incremental displacement jump vector $\mathbf{u}$ in the local boundary system as

$$\mathbf{u} = \{0, 0, \langle g_n \rangle\} + \mathbf{T}_b^{-T} \cdot \Delta \mathbf{g}_t$$

where $\mathbf{T}_b^{-T}$ represents the transformation from the global system to the boundary system. In the above expression, the Macaulay brackets indicate the positive parts operator such that

$$\langle x \rangle = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

Using the displacement jump vector, the adhesive stress vector $\mathbf{f}$ is defined as

$$\mathbf{f} = \mathbf{k}\mathbf{u}$$

where $\mathbf{k}$ is the adhesive stiffness. For negative values of g_n, the normal component of $\mathbf{f}$ is zero, and the contact condition is resolved by the penalty contact formulation. Notice that a different sign convention is used for the normal stress in the adhesion and contact contributions, where T_n is positive in compression.

The adhesive stiffness $\mathbf{k}$ can be defined using three different options: **From contact penalty factor**, **User defined**, and **Use material data**. For the first option, the normal stiffness is set equal to the contact pressure penalty factor p_n. The two tangential stiffness components are then assumed to be related to the normal stiffness, so that the stiffness vector equals

$$\mathbf{k} = \{n_\tau p_n, n_\tau p_n, p_n\}$$

where n_τ is a coefficient with the default value 0.17. This coefficient can either be input explicitly, or be computed from a Poisson's ratio. A plane strain assumption is used for this conversion, giving

$$n_\tau = \frac{1 - 2\nu}{2(1 - \nu)}$$

For the **Use material data** option, $\mathbf{k}$ is calculated from the elastic constants of a fictive layer with a thickness equal to d_s.

The adhesive contribution to the virtual work on the destination boundary is

$$\int_{\text{dst}} (\mathbf{f} \cdot \delta\mathbf{u}) dA$$

Decohesion

When adhesion is active, it is possible to break the bond between the source and destination boundaries by adding a Decohesion subnode to **Contact**. Decohesion modifies the stress vector $\mathbf{f}$ defined by **Adhesion**, but does not explicitly add any new contribution to the virtual work on the destination boundary. It thus requires an active **Adhesion** node.

Decohesion is implemented using a Cohesive Zone Model (CZM) based on interface damage mechanics. In essence, a CZM can be viewed as a regularization of linear elastic fracture mechanics, where the energy release rate is distributed over a process zone resulting in the progressive fracture of the interface. Here damage is assumed to be a scalar variable that affects both the normal and tangential components of the stress vector. Crack opening in the normal direction is called mode I fracture, while shear is called mode II (or mode III) fracture.

For a general CZM, the stress vector is defined as

$$\mathbf{f} = (1 - d)\mathbf{ku} = \mathbf{ku} - d\mathbf{ku} \tag{3-234}$$

where d is the damage variable. During crack opening (or shearing), the damage variable grows, resulting in a softening behavior of the interface until it eventually breaks when $d = 1$, see Figure 3-43. If the interface is unloaded, the material follows the linear secant stiffness as defined by the current state of damage. No permanent deformations remain at complete unloading.

The Decohesion subnode implements the second term on the right-hand side of Equation 3-234, while the first term is implemented in **Adhesion**. Notice that in the normal direction, damage only applies to separation of the boundaries, hence the normal contact is unaffected by decohesion.

Two different CZM are available, providing different alternatives on how to define the evolution of damage during decohesion.

DISPLACEMENT-BASED DAMAGE

In the displacement-based damage models, the damage variable d is defined using a damage evolution function written in terms of a displacement quantity. Since, in general, the fracture is a combination of mode I and mode II fracture, the model

introduces a mixed mode displacement u_{m} as the norm of the displacement jump vector.

$$u_{\mathrm{m}} = \|\mathbf{u}\|$$

To keep track of the current state of damage, the maximum value of u_{m} over the loading history is defined as

$$u_{\mathrm{m,\,max}} = \max(u_{\mathrm{m}}, u^{\mathrm{old}}_{\mathrm{m,\,max}})$$

where $u^{\mathrm{old}}_{\mathrm{m,\,max}}$ is and internal degree of freedom that takes the value of $u_{\mathrm{m,max}}$ at the previous converged solution. The damage variable is then defined as a function of $u_{\mathrm{m,max}}$ of the form

$$d = \begin{cases} 0 & \text{if } u_{\mathrm{m,\,max}} < u_{0\mathrm{m}} \\ \min[F^{-1}(u_{\mathrm{m,\,max}}), 1] & \text{else} \end{cases}$$

where F^{-1} is called the *damage evolution function* and $u_{0\mathrm{m}}$ defines the onset of damage. Conceptually, the damage evolution function is the inverse of the softening branch of the traction–separation law F. Four different definitions of the damage evolution function F^{-1} are available in the **Traction–separation law** list.

The damage evolution functions are summarized in Figure 3-42, and the resulting traction–separation laws in Figure 3-43.

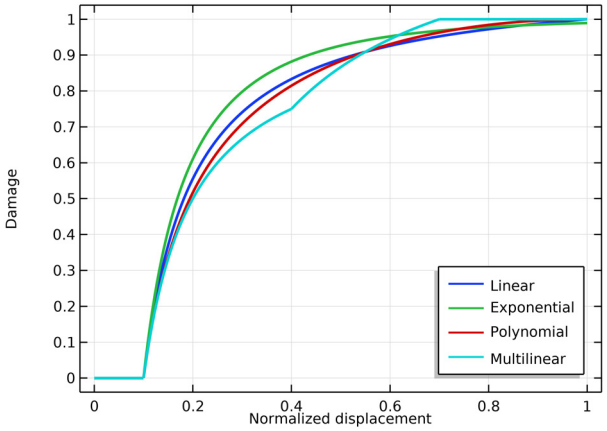


Figure 3-42: Damage evolution functions available with the displacement-based damage model.

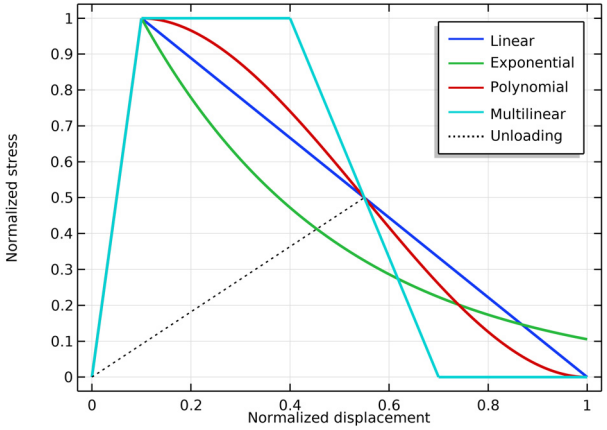


Figure 3-43: Traction–separation laws available with the displacement-based damage model.

The **Linear** option specifies a damage evolution function that gives a bilinear traction–separation law as seen in Figure 3-43. It is defined as

$$F^{-1}(u_{\mathrm{m,max}}) = \frac{u_{\mathrm{fm}}}{u_{\mathrm{m,max}}}\left(\frac{u_{\mathrm{m,max}} - u_{0\mathrm{m}}}{u_{\mathrm{fm}} - u_{0\mathrm{m}}}\right)$$

where $u_{0\mathrm{m}}$ and u_{fm} define the *mixed mode* initiation of damage and point of complete fracture, respectively. For the initiation of damage, a linear mixed mode criterion gives

$$u_{0\mathrm{m}} = u_{0\mathrm{t}} u_{0\mathrm{s}} \sqrt{\frac{u_{\mathrm{m}}^2}{\langle u_{\mathrm{I}} \rangle^2 u_{0\mathrm{s}}^2 + u_{\mathrm{II}}^2 u_{0\mathrm{t}}^2}} \tag{3-235}$$

where u_{I} and u_{II} are the mode I and mode II displacements, respectively. These are obtained from the displacement jump vector $\mathbf{u}$. The constants $u_{0\mathrm{t}}$ and $u_{0\mathrm{s}}$ are calculated as

$$u_{0\mathrm{t}} = \frac{\sigma_{\mathrm{ts}}}{k_{\mathrm{n}}} \quad \text{and} \quad u_{0\mathrm{s}} = \frac{\sigma_{\mathrm{ss}}}{k_{\mathrm{t}}} \tag{3-236}$$

where σ_{ts} is the tensile strength and σ_{ss} is the shear strength of the adhesive layer. The *normal stiffness* k_{n} and the equivalent *tangential stiffness* k_{t} are obtained from the adhesive stiffness vector $\mathbf{k}$. The mixed mode failure displacement u_{fm} depends on the selected **Mixed mode criterion**.

The **Power Law** criterion is defined as

$$\left(\frac{G_{\mathrm{I}}}{G_{\mathrm{ct}}}\right)^{\alpha} + \left(\frac{G_{\mathrm{II}}}{G_{\mathrm{cs}}}\right)^{\alpha} = 1$$

the **Generalized Power Law** criterion is defined as

$$\left(\frac{G_{\mathrm{I}}}{G_{\mathrm{ct}}}\right)^{\alpha_{\mathrm{t}}} + \left(\frac{G_{\mathrm{II}}}{G_{\mathrm{cs}}}\right)^{\alpha_{\mathrm{s}}} = 1$$

and the **Benzeggagh–Kenane** criterion is defined as

$$G_{\mathrm{ct}} + (G_{\mathrm{cs}} - G_{\mathrm{ct}})\left(\frac{G_{\mathrm{II}}}{G_{\mathrm{I}} + G_{\mathrm{II}}}\right)^{\alpha} = G_{\mathrm{cm}}$$

where G_{ct}, G_{cs}, and G_{cm} are the tensile, shear, and total energy release rates, respectively. The parameter α is called the *mode mixity exponent*. For the generalized power law criterion, the tensile and shear energy release rates have different mode mixity exponents, α_{t} and α_{s} respectively.

From these relations, the *mixed mode failure displacement* u_{fm} is calculated. For the power law criterion it reads

$$u_{\mathrm{fm}} = \begin{cases} \dfrac{2(1+\beta^2)}{u_{0\mathrm{m}}}\left[\left(\dfrac{k_{\mathrm{n}}}{G_{\mathrm{ct}}}\right)^{\alpha} + \left(\dfrac{\beta^2 k_{\mathrm{t}}}{G_{\mathrm{cs}}}\right)^{\alpha}\right]^{\frac{-1}{\alpha}} & \text{if } u_{\mathrm{I}} > 0 \\ \dfrac{2G_{\mathrm{cs}}}{\sigma_{\mathrm{s}}} & \text{if } u_{\mathrm{I}} \le 0 \end{cases} \tag{3-237}$$

where the *mode mixity ratio* reads $\beta = u_{\mathrm{II}}/u_{\mathrm{I}}$.

For the generalized power law criterion, the mixed mode failure displacement u_{fm} is obtained after solving the nonlinear equation

$$\left(\frac{k_{\mathrm{n}} u_{0\mathrm{m}} u_{\mathrm{fm}}}{2\sqrt{1+\beta^2}G_{\mathrm{ct}}}\right)^{\alpha_{\mathrm{t}}} + \left(\frac{\beta^2 k_{\mathrm{t}} u_{0\mathrm{m}} u_{\mathrm{fm}}}{2\sqrt{1+\beta^2}G_{\mathrm{cs}}}\right)^{\alpha_{\mathrm{s}}} = 1 \tag{3-238}$$

and for the Benzeggagh–Kenane criterion, the corresponding expression for the mixed mode failure displacement u_{fm} is

$$u_{\mathrm{fm}} = \begin{cases} \dfrac{2u_{\mathrm{m}}^2}{u_{0\mathrm{m}}(k_{\mathrm{n}}u_{\mathrm{I}}^2 + k_{\mathrm{t}}u_{\mathrm{II}}^2)}\left[G_{\mathrm{ct}} + (G_{\mathrm{cs}} - G_{\mathrm{ct}})\left(\dfrac{\frac{k_{\mathrm{t}}}{k_{\mathrm{n}}}u_{\mathrm{II}}^2}{u_{\mathrm{I}}^2 + \frac{k_{\mathrm{t}}}{k_{\mathrm{n}}}u_{\mathrm{II}}^2}\right)^{\alpha}\right] & \text{if } u_{\mathrm{I}} > 0 \\ \dfrac{2G_{\mathrm{cs}}}{\sigma_{\mathrm{s}}} & \text{if } u_{\mathrm{I}} \le 0 \end{cases} \tag{3-239}$$

The **Exponential** option specifies a damage evolution function that gives a traction–separation law which is linear up to the interface strength, and thereafter softens with an exponential curve that asymptotically reaches zero as shown in Figure 3-43,

$$F^{-1}(u_{\mathrm{m,max}}) = 1 - \frac{u_{0\mathrm{m}}}{u_{\mathrm{m,max}}}\exp\left(-\frac{u_{\mathrm{m,max}} - u_{0\mathrm{m}}}{u_{\mathrm{fm}} - u_{0\mathrm{m}}}\right)$$

The mixed mode damage initiation displacement $u_{0\mathrm{m}}$ is defined by Equation 3-235 and Equation 3-236.

The mixed mode failure displacement u_{fm} reads for the power law

$$u_{\mathrm{fm}} = \begin{cases} u_{0\mathrm{m}}(1+\beta^2)\left[\left(k_{\mathrm{n}}\frac{u_{0\mathrm{m}}^2}{G_{\mathrm{ct}}}\right)^{\alpha} + \left(k_{\mathrm{t}}\frac{\beta^2 u_{0\mathrm{m}}^2}{G_{\mathrm{cs}}}\right)^{\alpha}\right]^{\frac{-1}{\alpha}} & \text{if } u_{\mathrm{I}} > 0 \\ \frac{G_{\mathrm{cs}}}{\sigma_{\mathrm{s}}} & \text{if } u_{\mathrm{I}} \le 0 \end{cases}$$

For the generalized power law criterion, the mixed mode failure displacement u_{fm} is obtained after solving the nonlinear equation

$$\left(\frac{k_{\mathrm{n}} u_{0\mathrm{m}} u_{\mathrm{fm}}}{2\sqrt{1+\beta^2}G_{\mathrm{ct}}}\right)^{\alpha_{\mathrm{t}}} + \left(\frac{\beta^2 k_{\mathrm{t}} u_{0\mathrm{m}} u_{\mathrm{fm}}}{2\sqrt{1+\beta^2}G_{\mathrm{cs}}}\right)^{\alpha_{\mathrm{s}}} = 1$$

and for the Benzeggagh–Kenane criterion, the mixed mode failure displacement u_{fm} reads

$$u_{\mathrm{fm}} = \begin{cases} \frac{u_{\mathrm{m}}^2}{u_{0\mathrm{m}}(k_{\mathrm{n}}u_{\mathrm{I}}^2 + k_{\mathrm{t}}u_{\mathrm{II}}^2)}\left[G_{\mathrm{ct}} + (G_{\mathrm{cs}} - G_{\mathrm{ct}})\left(\frac{\frac{k_{\mathrm{t}}}{k_{\mathrm{n}}}u_{\mathrm{II}}^2}{u_{\mathrm{I}}^2 + \frac{k_{\mathrm{t}}}{k_{\mathrm{n}}}u_{\mathrm{II}}^2}\right)^{\alpha}\right] & \text{if } u_{\mathrm{I}} > 0 \\ \frac{G_{\mathrm{cs}}}{\sigma_{\mathrm{s}}} & \text{if } u_{\mathrm{I}} \le 0 \end{cases}$$

The **Polynomial** option specifies a damage evolution function that gives a linear traction–separation law up to the interface strength, it thereafter softens with a cubic polynomial curve as shown in Figure 3-43,

$$F^{-1}(u_{\mathrm{m,max}}) = 1 - \frac{u_{0\mathrm{m}}}{u_{\mathrm{m,max}}}\left[1 + \left(\frac{u_{\mathrm{m,max}} - u_{0\mathrm{m}}}{u_{\mathrm{fm}} - u_{0\mathrm{m}}}\right)^2\left(2\frac{u_{\mathrm{m,max}} - u_{0\mathrm{m}}}{u_{\mathrm{fm}} - u_{0\mathrm{m}}} - 3\right)\right]$$

The mixed mode damage initiation displacement $u_{0\mathrm{m}}$ is defined by Equation 3-235 and Equation 3-236, and the mixed mode failure displacement u_{fm} by Equation 3-237, Equation 3-238, or Equation 3-239.

The **Multilinear** option specifies a damage evolution function that gives a traction–separation law that is linear up to the interface strength. Thereafter a region of constant stress is introduced before the interface softens linearly as seen in Figure 3-43.

$$F^{-1}(u_{\mathrm{m,\,max}}) = \begin{cases} 1 - \dfrac{u_{0\mathrm{m}}}{u_{\mathrm{m,\,max}}} & \text{if } u_{\mathrm{m,\,max}} < u_{\mathrm{pm}} \\ 1 - \dfrac{u_{0\mathrm{m}}}{u_{\mathrm{m,\,max}}}\left(\dfrac{u_{\mathrm{fm}} - u_{\mathrm{m,\,max}}}{u_{\mathrm{fm}} - u_{\mathrm{pm}}}\right) & \text{if } u_{\mathrm{m,\,max}} \geq u_{\mathrm{pm}} \end{cases}$$

The mixed mode damage initiation displacement $u_{0\mathrm{m}}$ is defined by Equation 3-235 and Equation 3-236.

The variable u_{pm} defines the end of the region of constant stress and introduces the *shape factor* λ. The shape factor defines the ratio between the constant stress part of G_{ct} and the total "inelastic" part of G_{ct}:

$$\lambda = \frac{\sigma_{\mathrm{t}}(u_{\mathrm{pt}} - u_{0\mathrm{t}})}{\sigma_{\mathrm{t}}(u_{\mathrm{pt}} - u_{0\mathrm{t}}) + \frac{1}{2}\sigma_{\mathrm{t}}(u_{\mathrm{ft}} - u_{\mathrm{pt}})} = \frac{\sigma_{\mathrm{t}}(u_{\mathrm{pt}} - u_{0\mathrm{t}})}{G_{\mathrm{ct}} - \frac{1}{2}\sigma_{\mathrm{t}}u_{0\mathrm{t}}}$$

Note that the shape factor is similarly defined for shear and it is assumed to be equal for both components. Setting $\lambda = 0$ corresponds to the linear separation law.

Using the above expression, the stress plateau displacement, $u_{\mathrm{p}i}$, for the respective component can be expressed as

$$u_{\mathrm{p}i} = \lambda\frac{G_{\mathrm{c}i}}{\sigma_i} + u_{0i}\left(1 - \frac{\lambda}{2}\right)$$

and the failure displacement $u_{\mathrm{f}i}$ as

$$u_{\mathrm{f}i} = (2 - \lambda)\frac{G_{\mathrm{c}i}}{\sigma_i} + \frac{\lambda u_{0i}}{2}$$

where the index i indicates either tension or shear.

For the multilinear option, the mixed mode criterion is always linear ($\alpha = 1$). Hence the mixed mode stress plateau displacement, u_{pm}, and the failure displacement, u_{fm}, are given as

$$u_{\mathrm{pm}} = \sqrt{\frac{u_{\mathrm{m,\,max}}^2}{u_{\mathrm{I}}^2 u_{\mathrm{ps}}^2 + u_{\mathrm{II}}^2 u_{\mathrm{pt}}^2}}$$

$$u_{\mathrm{fm}} = 2G_{\mathrm{ct}}G_{\mathrm{cs}}\frac{1+\beta^2}{u_{0\mathrm{m}}(k_{\mathrm{n}}G_{\mathrm{cs}}+\beta^2 k_{\mathrm{t}}G_{\mathrm{cs}})} - (u_{\mathrm{pm}} - u_{0\mathrm{m}})$$

ENERGY-BASED DAMAGE

The energy-based damage models relate the growth of damage to the dissipated mechanical energy of the interface. The formulation is based on the work presented in Ref. 165 and Ref. 166. The derivation starts from the stored energy function

$$\psi(\mathbf{u}, d) = \frac{1}{2}(1-d)\mathbf{k}\mathbf{u}\cdot\mathbf{u}$$

From the stored energy $\psi(\mathbf{u}, d)$, the stress vector $\mathbf{f}$ and damage energy release rate Y_{dm} are obtained as

$$\mathbf{f} = \frac{\partial(\psi(\mathbf{u}, d))}{\partial \mathbf{u}} = (1-d)\mathbf{k}\mathbf{u}$$

$$Y_{\mathrm{dm}} = -\frac{\partial(\psi(\mathbf{u}, d))}{\partial d} = \frac{1}{2}\mathbf{k}\mathbf{u}\cdot\mathbf{u}$$

To keep track of the current state of damage, the maximum value of Y_{d} over the loading history is defined as

$$Y_{\mathrm{dm,\,max}} = \max(Y_{\mathrm{dm}}, Y^{\mathrm{old}}_{\mathrm{dm,\,max}})$$

where $Y^{\mathrm{old}}_{\mathrm{d,\,max}}$ is an internal variable that takes the value of $Y_{\mathrm{dm,max}}$ at the previous converged solution. The energy dissipated during the decohesion process is

$$\int_0^{\infty} \dot{Y}_{\mathrm{dm,\,max}}\,dt = G_{\mathrm{cm}}$$

where G_{cm} is the critical energy release rate in the sense of fracture mechanics. The overall behavior of the cohesive zone model is then summarized by

$$Y_{\mathrm{dm,\,max}} = \begin{cases} Y_{\mathrm{d0m}} & \text{if } d = 0 \\ \int_0^t \dot{Y}_{\mathrm{dm,\,max}}\,dt = F(d) & \text{if } 0 < d < 1 \\ Y_{\mathrm{dm}} & \text{if } d = 1 \end{cases}$$

where Y_{d0m} defines the damage threshold, and $F(d)$ is a monotonically increasing function of the damage variable. From the above, an expression for the damage variable is obtained as

$$d = \begin{cases} 0 & Y_{\mathrm{dm,\,max}} < Y_{\mathrm{dm0}} \\ \min[F^{-1}(Y_{\mathrm{dm,\,max}}),\, 1] & \text{else} \end{cases}$$

Different definitions for F^{-1} are available under the **Traction–separation law** list. These damage evolution functions are summarized in Figure 3-44, and the resulting traction–separation laws in Figure 3-45.

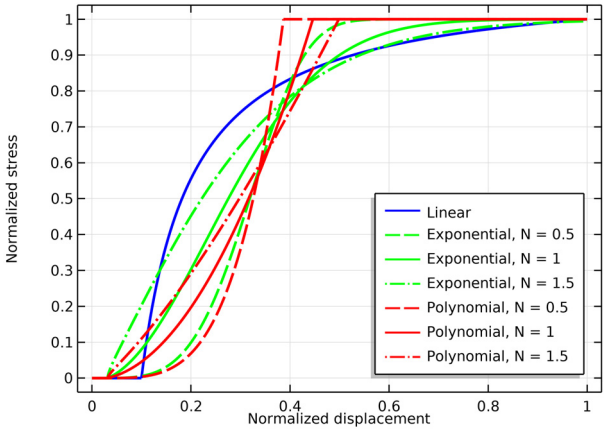


Figure 3-44: Damage evolution functions available with the energy-based damage model. Note that the initial damage threshold is 10 times lower for the Exponential and Polynomial laws.

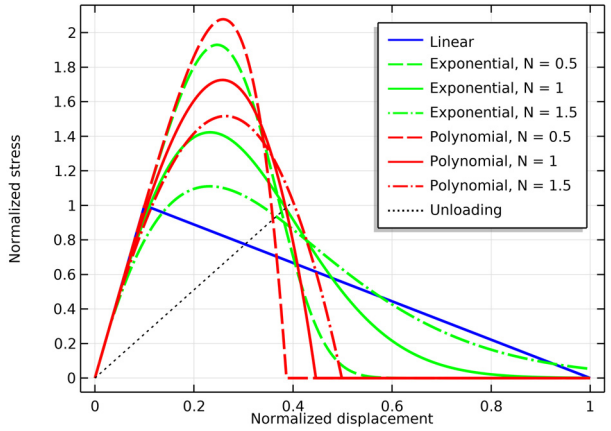


Figure 3-45: Traction–separation laws available with the energy-based damage model. Note that the initial damage threshold is 10 times lower for the Exponential and Polynomial laws.

The **Linear** option specifies a damage evolution function that gives a bilinear traction–separation law as shown in Figure 3-45.

$$F^{-1}(Y_{\mathrm{d,\,max}}) = \frac{Y_{\mathrm{dfm}}}{Y_{\mathrm{d0m}} - Y_{\mathrm{dfm}}}\left(\sqrt{\frac{Y_{\mathrm{d0m}}}{Y_{\mathrm{d,\,max}}}} - 1\right)$$

For the linear law, the equation constant $Y_{\mathrm{dfm}} = G_{\mathrm{cm}}$.

The **Exponential** option specifies a more general damage evolution function of the form

$$F^{-1}(Y_{\mathrm{d,\,max}}) = 1 - \exp\left(\frac{Y_{\mathrm{d,\,max}} - Y_{\mathrm{d0m}}}{Y_{\mathrm{dfm}} - Y_{\mathrm{d0m}}}\right)^{\frac{1}{N}}$$

where N is a smoothing parameter with a default value equal to 1. The effect of N on the traction–separation curve can be seen in Figure 3-45. For the exponential law, Y_{dfm} is defined as

$$Y_{\mathrm{dfm}} = Y_{\mathrm{d0m}} + \frac{G_{\mathrm{cm}} - Y_{\mathrm{d0m}}}{\Gamma(1+N)}$$

where $\Gamma()$ is the gamma function.

A similar damage evolution function is obtained with the **Polynomial** option. It gives a generalized version of the traction–separation law proposed in Ref. 166,

$$F^{-1}(Y_{\mathrm{d,\,max}}) = \left(\frac{Y_{\mathrm{d,\,max}} - Y_{\mathrm{d0m}}}{Y_{\mathrm{dfm}} - Y_{\mathrm{d0m}}}\right)^{\frac{1}{N}}$$

where

$$Y_{\mathrm{dfm}} = N\left[\left(\frac{1}{N} + 1\right)G_{\mathrm{cm}} - Y_{\mathrm{d0m}}\right]$$

The definition of the two variables, Y_{d0m} and G_{cm}, requires the introduction of a few concepts related to the mixed mode loading. First, a *mixed mode ratio* is introduced

$$\beta = \sqrt{\frac{k_{\mathrm{t}}}{k_{\mathrm{n}}}}\left(\frac{u_{\mathrm{II}}}{u_{\mathrm{I}}}\right)$$

where u_{I} and u_{II} are the mode I and mode II displacements, respectively. These are obtained from the displacement jump vector **u**.

The energy release rate of the respective modes is then given by

$$Y_{\mathrm{dI}} = \frac{1}{1+\beta^2}Y_{\mathrm{dm}} \text{ and } Y_{\mathrm{dII}} = \frac{\beta^2}{1+\beta^2}Y_{\mathrm{dm}}$$

Two mode mixture rules are formulated, one for the initiation of damage and one for the critical energy release rates. A power law criterion is assumed for both, such that for the onset of damage

$$\left(\frac{Y_{\mathrm{dI}}}{G_{0\mathrm{t}}}\right)^{\alpha_0} + \left(\frac{Y_{\mathrm{dII}}}{G_{0\mathrm{s}}}\right)^{\alpha_0} = 1$$

where $G_{0\mathrm{t}}$ and $G_{0\mathrm{s}}$ define the damage threshold in tension and shear, respectively.

The corresponding criterion for the fracture toughness is

$$\left(\frac{G_{\mathrm{I}}}{G_{\mathrm{ct}}}\right)^{\alpha_{\mathrm{c}}} + \left(\frac{G_{\mathrm{II}}}{G_{\mathrm{cs}}}\right)^{\alpha_{\mathrm{c}}} = 1$$

where G_{ct} and G_{cs} are the critical energy release rates for tension and shear, respectively. The variables α_0 and α_{c} are called the *mode mixity exponents.*

Using these mode mixture rules results in

$$Y_{\mathrm{d0m}} = \frac{(1+\beta^2)G_{0\mathrm{t}}G_{0\mathrm{s}}}{\left\{G_{0\mathrm{s}}^{\alpha_0} + (\beta^2 G_{0\mathrm{t}})^{\alpha_0}\right\}^{\frac{1}{\alpha_0}}}$$

$$G_{\mathrm{cm}} = \frac{(1+\beta^2)G_{\mathrm{ct}}G_{\mathrm{cs}}}{\left\{G_{\mathrm{cs}}^{\alpha_c} + (\beta^2 G_{\mathrm{ct}})^{\alpha_c}\right\}^{\frac{1}{\alpha_c}}}$$

REGULARIZATION

Due to the release of energy during decohesion and the softening behavior of the damage models, models may suffer from unwanted mesh dependency. The CZM can suffer from instabilities that lead to nonconvergent solutions, or bifurcations resulting into different solutions for different meshes.

Such deficiencies may be alleviated by introducing some additional form of regularization. Under the **Regularization** list, it is possible to add the **Delayed damage** regularization. This option, available for time-dependent studies, adds a viscous delay to the damage growth. The formulation introduces a viscous damage variable d_{v}. It redefines Equation 3-234 so that

$$\mathbf{f} = (1-d_{\mathrm{v}})\mathbf{k}^{\mathrm{T}}\mathbf{u} = \mathbf{k}^{\mathrm{T}}\mathbf{u} - d_{\mathrm{v}}\mathbf{k}^{\mathrm{T}}\mathbf{u}$$

The variation of the viscous damage variable is defined as

$$\dot{d}_{\mathrm{v}} = -\frac{1}{\tau}(d_{\mathrm{v}} - d) \tag{3-240}$$

where d is the damage variable obtained from any of the available CZM, and τ is the characteristic time that defines the delay of the decohesion. If the viscous damage is used to stabilize a rate-independent decohesion problem, the value of τ must be chosen with care. As a rule of thumb, τ should at least be one or two orders of magnitude smaller than the expected time step. Values of τ that are too large can introduce significant amounts of extra fracture energy to the model, and the actual energy dissipated due to damage can exceed the defined critical energy release rates by orders of magnitude.

The current value of the viscous damage variable, d_v^{n+1}, is obtained after numerical integration of Equation 3-240 such that

$$d_v^{n+1} = \exp\left(-\frac{\Delta t}{\tau}\right) d_v^n + \left[1 - \exp\left(-\frac{\Delta t}{\tau}\right)\right] d^{n+1}$$

where Δt is the current time step taken by the time-dependent solver. The value of the viscous damage at the previously converged step, d_v^n, is stored as an internal variable.

DAMAGE DISSIPATION

The dissipation due to damage is given as

$$\dot{W}_{dmg} = \dot{d} W_{sd} = \dot{d}\left(\frac{1}{2}\mathbf{k}^T\mathbf{u} \cdot \mathbf{u}\right)$$

The energy dissipated due to damage at the current step is approximated by

$$W_{dmg}^{n+1} = (d^{n+1} - d^n)\left(\frac{W_{sd}^{n+1} - W_{sd}^{n+1}}{2} + W_{sd}^{n+1}\right) + W_{dmg}^n$$

where the required values at the previously converged step n are stored as internal variables.

Wear

By adding a Wear subnode to a **Contact** node, it is possible to model adhesive or abrasive wear of the material when the contacting boundaries are sliding along each other. Since wear involves solving evolution equations, the **Wear** node only adds a contribution for time-dependent studies.

In the Solid Mechanics and the Multibody Dynamics interfaces, the removal of material during the wear process can be modeled with two fundamentally different techniques. The most general technique is to model the removal using the deformed geometry concept. With this approach, the material frame $\mathbf{X}$ of the domains adjacent to the contacting boundaries is updated according to the computed wear depth h_{wear}. This means that there is an actual removal of material during the simulation which affects, for example, the contact search, mapping, and conditions. When selecting the **Deformed geometry** formulation, the wear feature adds a (hidden) Deforming Domain feature that controls the material frame through an adaptive mesh smoothing. The removal of material is made through a (hidden) Prescribed Normal Mesh Displacement boundary condition controlled by h_{wear} on the selected contact

boundaries. By adding the deformed geometry, an extra dependent variable is added, the material mesh displacement `material.disp`. This adds a set of extra degrees of freedom to the model that needs to be solved for.

Alternatively, the removal of material can be modeled using an offset-based approach. This formulation offers a simplified approach that is computationally less expensive, but mainly suitable when the amount of worn-off material is small. When selecting the **Offset-based** formulation, h_{wear} is subtracted from the offset variables $d_{\text{offset,s}}$ and $d_{\text{offset,d}}$ in Equation 3-217. Hence, the material is considered removed only in the definition of the physical gap, while the contact search and mapping are unaffected. The latter follows from the fact that the actual coordinates and normals of the contacting boundaries essentially remain constant with respect to the wear; they, however, can change due to the deformation induced by the wear and changing contact conditions.

In the Shell and Membrane interfaces, wear is modeled by simultaneously changing the thickness variable and the offset from the midsurface to the meshed boundary. With this approach wear can also be modeled on the top side, the bottom side, or on both sides at the same time.

The accumulated wear is computed from an evolution equation of the following general format

$$\frac{\partial h_{\text{wear}}}{\partial t} = f\left(\mathbf{v}_{\text{slip}}, T_{\text{N}}, T, \theta\right)$$

where the rate of the wear depth is given by some source term f that is typically a function of the slip velocity $\mathbf{v}_{\text{slip}}$, the contact pressure T_{n}, and the temperature T. The surface and material properties also play an important role, and are represented by the generic quantity θ in the above equation.

In COMSOL Multiphysics, the wear depth can be computed using a **Generalized Archard** law for which the wear rate is

$$\frac{\partial h_{\text{wear}}}{\partial t} = k_{\text{wear}}\left(\frac{T_{\text{n}}}{T_{\text{n,ref}}}\right)^{n}\left\|\mathbf{v}_{\text{slip}}\right\| \tag{3-241}$$

Here k_{wear} is a dimensionless wear constant and the exponent n controls the dependence of the wear rate on the contact pressure. The reference contact pressure, $T_{\text{n,ref}}$, can be chosen arbitrarily, and is used only to obtain consistent units. The classical Archard wear equation is retrieved from Equation 3-241 by setting $n = 1$ and

$T_{\mathrm{n,ref}} = 1$ Pa. In addition, it is also possible to enter an arbitrary expression for the source term f that defines the wear rate.

It is possible to account for wear on both the source and destination boundaries. However, it is generally more accurate to model wear on the destination side. This follows from the fact that most relevant quantities, such as T_{n} and $\mathbf{v}_{\mathrm{slip}}$, are defined only on the destination boundary in the **Contact** node and by, for example, the **Friction** node. Hence, when modeling wear on the source boundary, these quantities are mapped from the destination to the source. For example, when applied to the source side, Equation 3-241 actually reads

$$\frac{\partial h_{\mathrm{wear}}}{\partial t} = k_{\mathrm{wear}}\left[\frac{\mathrm{map}(T_{\mathrm{n}}, \mathbf{x}_{\mathrm{src}})}{T_{\mathrm{n,ref}}}\right]^{n} \mathrm{map}(\|\mathbf{v}_{\mathrm{slip}}\|, \mathbf{x}_{\mathrm{src}})$$

The definition of the wear rate thus includes multiple mappings form source to destination, and form destination to source. As described in Contact Search and Kinematics, these mappings are not necessarily one-to-one, which can lead to accumulating errors. The offset-based wear formulation can be especially sensitive to such errors.

Bolt Modeling Theory

Tightening Torque

When the bolt preload is given in terms of the tightening torque, a detailed mechanical analysis of the bolt mechanics is needed for the conversion.

The total required torque, M_T, is to a large extent determined by frictional losses. It can, however, be shown that it is directly proportional to the axial force in the bolt, F.

The torque consists of a sum of two parts, the torque in the thread, M_t, and the torque caused by friction under the head, M_h,

$$M_T = M_t + M_h$$

The moment around the bolt axis caused by rotating the bolt head can be expressed as

$$M_h = \mu_h F r_e$$

Here, μ_h is the coefficient of friction under the bolt head (or nut, whichever is rotating), and r_e is an effective radius at which the circumferential force can be considered as acting. If the contact pressure under the bolt head is assumed to be constant, then the effective radius can be computed as

$$r_e = \frac{\int r dA}{\int dA}$$

Here, r is the distance from the bolt axis, and the integrals are taken over the contact area under the bolt head.

For a circular bolt head, this gives

$$r_e = \frac{1}{3}\frac{d_o^3 - d_h^3}{d_o^2 - d_h^2}$$

where d_o is the outer diameter of the bolt head, and d_h is the bolt hole diameter.

To a first-order approximation, the effective radius is

$$r_e \approx \frac{d_o + d_h}{4}$$

For a hexagonal bolt head, the expression for the effective radius is

$$r_e = \frac{\left(\frac{1}{3} + \frac{\ln(3)}{4}\right)H^3 - \frac{\pi}{6}d_h^3}{\sqrt{3}H^3 - \frac{\pi}{2}d_h^2}$$

Here, H is the width (spanner size) of the bolt.

For a wide range of bolt configurations, it is reasonable to do the following approximations in terms of the nominal bolt diameter, d:

$d_h \approx 1{,}1d$

$d_o = H \approx 1{,}6d$

giving

$r_e \approx 0{,}68d$ for a circular bolt head

$r_e \approx 0{,}71d$ for a hexagonal bolt head

To compute the bolt thread moment, a detailed analysis of the thread geometry is necessary. Omitting the details,

$$M_t = \frac{F d_p \tan(\phi + \varepsilon)}{2}$$

Here, d_p is the mean diameter of the bolt thread.

For a wide range of bolt sizes, the bolt thread diameter can be approximated by $d_p \approx 0{,}9d$.

The lead angle of the bolt, ϕ, is given by

$$\tan(\phi) = \frac{l}{\pi d_p}$$

where l is the lead (expressed in the same units as d_p). This term, which is purely geometrical, gives the relation between torque and bolt force under ideal (frictionless) conditions.

The friction angle ε is defined as

$$\tan(\varepsilon) = \frac{\mu_\mathrm{t}}{\cos(\phi)\cos(\alpha)}$$

Here, μ_t is the coefficient of friction in the thread, while α is half the thread angle.

Since all terms in the torque are proportional to the bolt force, the axial force in the bolt can now be computed from the torque as

$$F = \frac{M_\mathrm{T}}{\mu_\mathrm{h} r_\mathrm{e} + \frac{d_p \tan(\phi + \varepsilon)}{2}}$$

For many threads, for example in the ISO and UTS standards, $\alpha = 30°$. Since the lead angle ϕ is small (about 3°), a good approximation is $\tan(\varepsilon) \approx 1{,}15\mu_\mathrm{t}$.

The sum of tangents formula gives

$$\tan(\phi + \varepsilon) = \frac{\tan(\phi) + \tan(\varepsilon)}{1 - \tan(\phi)\tan(\varepsilon)} \approx \tan(\phi) + \tan(\varepsilon)$$

Using the estimates above,

$$F \approx \frac{M_\mathrm{T}}{d(0{,}7\mu_\mathrm{h} + 0{,}03 + 0{,}5\mu_\mathrm{t})}$$

Thus, for ordinary values of the friction, 80 – 90% of the applied torque is consumed by the friction losses.

Theory for Spot Welds

Introduction

In this section, the theory for the Spot Welds node in the Shell interface is presented.

Spot welds are modeled using a mesh independent approach. A spot weld is defined by its location on the *destination* boundary. This is the boundary on which the equations are formed. The connection is created inside a circle defined by the *nugget diameter*.

The location of the spot weld does not have to be exactly on the destination boundary. The given point is projected to the nearest location on the destination boundary.

Stiffness

Essentially, the spot weld is modeled using a thin elastic layer. The stiffness of the elastic layer is separated into a normal stiffness, k_n, and a shear stiffness, k_s. The stiffness can be given by the user. The default is that it is automatically computed using the following expressions.

$$k_n = \frac{E_{equ}(1-\nu_{equ})}{(1+\nu_{equ})(1-2\nu_{equ})L_{cohe}}$$

$$k_s = \frac{E_{equ}}{2(1+\nu_{equ})L_{cohe}}$$

Here, E_{equ} and ν_{equ} are the equivalent Young's modulus and Poisson's ratio values respectively. The values are taken on the destination side. L_{cohe} is a parameter called the *cohesion length*. It represents the thickness of the virtual thin elastic layer, and should this be of the order of the distance between the welded plates. The default value is 1% of the nugget diameter.

The connection is made using numerical integration on the destination side. In order to get a good result, the element size should not be larger than the nugget size, and preferable somewhat smaller. By increasing the numerical integration order, you can, to some extent offset the effects of a coarse mesh.

For each integration point, a search is made to find the corresponding location on the *source* boundary. Thus, a displacement difference can be found:

$$\Delta\mathbf{u} = \mathbf{u}_{\mathrm{src}} - \mathbf{u}_{\mathrm{dst}}$$

Using the normal to the shell on the destination side, $\mathbf{n}$, the displacement difference can be separated in normal and tangential parts:

$$\Delta u_{\mathrm{n}} = \Delta\mathbf{u} \cdot \mathbf{n}$$

$$\Delta\mathbf{u}_{\mathrm{t}} = \Delta\mathbf{u} - \Delta u_{\mathrm{n}}\mathbf{n}$$

The contribution to the virtual work for the nugget is then

$$\delta W = \int_A [k_{\mathrm{n}}\Delta u_{\mathrm{n}}\mathrm{test}(\Delta u_{\mathrm{n}}) + k_{\mathrm{s}}\Delta\mathbf{u}_{\mathrm{t}} \cdot \mathrm{test}(\Delta\mathbf{u}_{\mathrm{t}})]dA$$

Force Evaluation

For each nugget, a set of force and moment variables are created. The underlying assumption is that there is a stress distribution in the thin elastic layer which is proportional to the difference in displacements between the two boundaries. The traction vector in global coordinates is

$$\mathbf{t} = k_{\mathrm{n}}\Delta u_{\mathrm{n}}\mathbf{n} + k_{\mathrm{s}}\Delta\mathbf{u}_{\mathrm{t}}$$

The force vector in global coordinates is obtained by integration over the nugget area,

$$\mathbf{F} = \int_A \mathbf{t}dA$$

Finally, the force is decomposed in a normal force and a shear force:

$$F_{\mathrm{n}} = \mathbf{F} \cdot \mathbf{n}$$
$$F_{\mathrm{s}} = \sqrt{(\mathbf{F} - F_{\mathrm{n}}\mathbf{n}) \cdot (\mathbf{F} - F_{\mathrm{n}}\mathbf{n})}$$

The moments are evaluated around the center of the nugget ($\mathbf{X}_{\mathrm{n}}$), that is the location of the spot weld after projection on the destination boundary. In global directions, the moment vector $\mathbf{M}$ is

$$\mathbf{M} = \int_A \mathbf{t} \times (\mathbf{X} - \mathbf{X}_{\mathrm{n}})dA$$

In the same way as the force, the moment is decomposed in a twisting moment around the normal of the boundary and a bending moment.

$$M_t = \mathbf{M} \cdot \mathbf{n}$$
$$M_b = \sqrt{(\mathbf{M} - M_t\mathbf{n}) \cdot (\mathbf{M} - M_t\mathbf{n})}$$

Theory for Fasteners

Introduction

In this section, the theory for the Fasteners node is presented. In the Shell physics interface the theory is valid for rivets, but with some assumptions it can also be applied to bolts. The assumptions are discussed in more detail in Simplified Modeling of Fasteners.

Stiffness

The connection between the two boundaries is modeled using a spring matrix with stiffness properties designed to replicate the flexibility introduced when two holes are connected by a fastener. The displacements at the two ends of the fastener are represented by the average displacement of each corresponding hole.

For the Shell interface, the displacement is taken on the plate side defined by a *connected location*. This side can be determined automatically or specified by the user. In either case, the key assumption is that there is no offset in the direction of the shell normal between the measurement points. In other words:

- The connection is set up so that the actual surfaces (not necessarily the mesh) of the two plates that are joined are adjacent.
- The connection is assumed to transmit no moment, but only translational forces.

The stiffness in the normal direction, k_{n}, is a given by the axial stiffness of a bar

$$k_{\mathrm{n}} = \frac{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}{4(t_{\mathrm{dst}} + t_{\mathrm{src}})}$$

Here, E_{f} is the fastener Young's modulus and d_{f} is the fastener diameter. The total length is the sum of the thickness of the plate on the destination side, t_{dst}, and the thickness on the source side, t_{src}.

In the Solid Mechanics interface, the **Fastener** node can be used to model a pretensioned bolt connection. To account for this, the effective length of the fastener includes an estimated bolt head thickness. The normal stiffness then becomes

$$k_{\mathrm{n}} = \frac{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}{4\left(t_{\mathrm{dst}} + t_{\mathrm{src}} + \frac{9}{10} d_{\mathrm{f}}\right)}$$

The stiffness in the transverse direction, k_{s}, is more complicated. There are several contributions to the deformation of the rivet-plate system. Four compliance terms connected in series are considered,

$$k_{\mathrm{s}} = \frac{1}{C_{\mathrm{b}} + C_{\mathrm{s}} + C_{\mathrm{br}} + C_{\mathrm{pbr}}}$$

Here, C_{b} is the fastener bending flexibility, C_{s} is the fastener shear flexibility, C_{br} is the fastener bearing flexibility, and C_{pbr} is the plate bearing flexibility.

The compliance contributions are defined as

$$C_{\mathrm{b}} = \alpha_{\mathrm{b}} \frac{3t_{\mathrm{src}}^3 + 16t_{\mathrm{src}}^2 d_{\mathrm{dst}} + 16t_{\mathrm{dst}}^2 d_{\mathrm{src}} + 3t_{\mathrm{dst}}^3}{2\pi E_{\mathrm{f}} d_{\mathrm{f}}^4}$$

$$C_{\mathrm{s}} = \alpha_{\mathrm{s}} \frac{7 + 14\nu_{\mathrm{f}} + 8\nu_{\mathrm{f}}^2}{2(1 + \nu_{\mathrm{f}})} \cdot \frac{t_{\mathrm{dst}} + t_{\mathrm{src}}}{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}$$

$$C_{\mathrm{br}} = \alpha_{\mathrm{br}} \left(\frac{1}{t_{\mathrm{src}} E_{\mathrm{f}}} + \frac{1}{t_{\mathrm{dst}} E_{\mathrm{f}}} \right)$$

$$C_{\mathrm{pbr}} = \alpha_{\mathrm{pbr}} \left(\frac{1}{t_{\mathrm{src}} E_{\mathrm{src}}} + \frac{1}{t_{\mathrm{dst}} E_{\mathrm{dst}}} \right)$$

In these expressions

- E_{src} and E_{dst} are the elastic moduli of the shell on the source and destination sides, respectively.
- E_{f}, ν_{f}, and d_{f} are the fastener stiffness, Poisson's ratio, and diameter, respectively.
- t_{src} and t_{dst} are the thickness of the plate on the source and destination sides, respectively.
- The multipliers α_m make it possible to adjust the analytical expressions.

In case of a pretension load, the clamped plates transfer a large part of the applied load. This partitioning of the load is a function of the relative stiffness of the bolt and the joint members. With an increment in the external load, ΔF_{ext}, then

$$\Delta F_{\text{ext}} = \Delta F_{\text{n}} + \Delta F_{\text{g}}$$
$$\Delta F_{\text{n}} = C_{\text{j}} \Delta F_{\text{ext}}$$
$$\Delta F_{\text{g}} = (1 - C_{\text{j}}) \Delta F_{\text{ext}}$$

Here, ΔF_{n} is the change in load carried by the bolt, and ΔF_{g} is the change in the load carried by the plates. The joint constant, C_{j}, is defined as

$$C_{\text{j}} = \frac{k_{\text{n}}}{k_{\text{g}} + k_{\text{n}}}$$

where k_{g} is the called *grip stiffness*, explained below.

As opposed to when using a full bolt pretension study, the fasteners offer a simplified approach where there is no special study step for computing the preloaded case. Rather, an analytical approximation to the state after preload is used.

In the following, $\mathbf{u}_{\text{n}}$ is the displacement in the normal direction between the plates, measured positive when the fastener is extended, so that the increment in the fastener force is

$$\Delta F_{\text{n}} = k_{\text{n}} \Delta \mathbf{u}_{\text{n}}$$

Now, assume a preload, F_{pre}. After pretensioning the bolt, there is no resulting external force. The preload in the fastener is balanced by the stress in the surrounding plates. That is

$$0 = F_{\text{n}} + F_{\text{g}}$$
$$F_{\text{n}} = F_{\text{pre}}$$

The connection will be compressed a distance $\mathbf{u}_{\text{n0}}$, which is negative. The force in the joint members is compressive,

$$F_{\text{g}} = k_{\text{g}} \mathbf{u}_{\text{n0}} = -F_{\text{pre}}$$

which provides a value for $\Delta \mathbf{u}_{\text{n0}}$. It is now possible to write the expression for the total fastener force under an arbitrary extension as

$$F_{\text{n}} = F_{\text{pre}} + k_{\text{n}}(\mathbf{u}_{\text{n}} - \mathbf{u}_{\text{n0}}) = F_{\text{pre}}\left(1 + \frac{k_{\text{n}}}{k_{\text{g}}}\right) + k_{\text{n}} \mathbf{u}_{\text{n}}$$

The fastener normal force can thus be expressed in a compact way as

$$F_{\mathrm{n}} = \frac{F_{\mathrm{pre}}}{1 - C_{\mathrm{j}}} + k_{\mathrm{n}} \Delta \mathbf{u}_{\mathrm{n}}$$

To approximate the grip stiffness, the following approach is used. The pressure distribution through the thickness of the grip is approximated by a conical shape, also called *frustum*. The total grip stiffness is a combination of the frustum stiffnesses as described below

$$\frac{1}{k_{\mathrm{g}}} = \sum_i \frac{1}{k_i}$$

When the connected plates have the same thickness, two frustums are considered as shown in the figure below

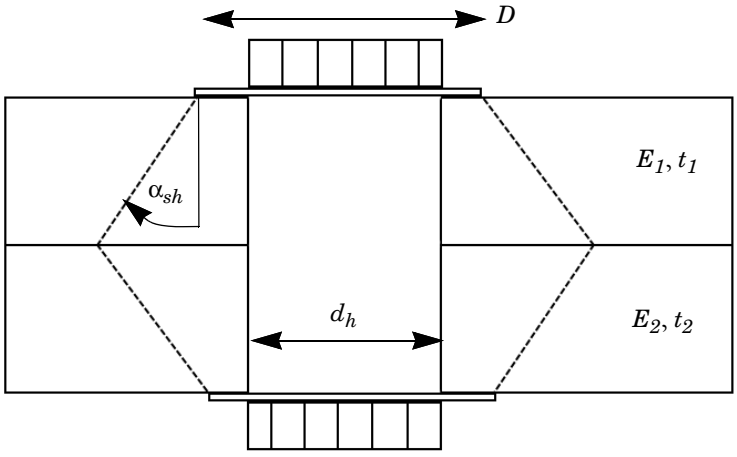


Figure 3-46: Conical pressure profile for two plates of the same thickness.

For each frustum, the stiffness is estimated using Shigley's method as expressed by

$$k_i = \frac{\pi E_i \tan \alpha_{sh}}{\log\left(\frac{2t_i \tan \alpha_{sh} + D - d_h}{2t_i \tan \alpha_{sh} + D + d_h} \cdot \frac{D + d_h}{D - d_h}\right)}$$

When the plates do not have the same thickness, a third frustum is introduced as shown in the figure below

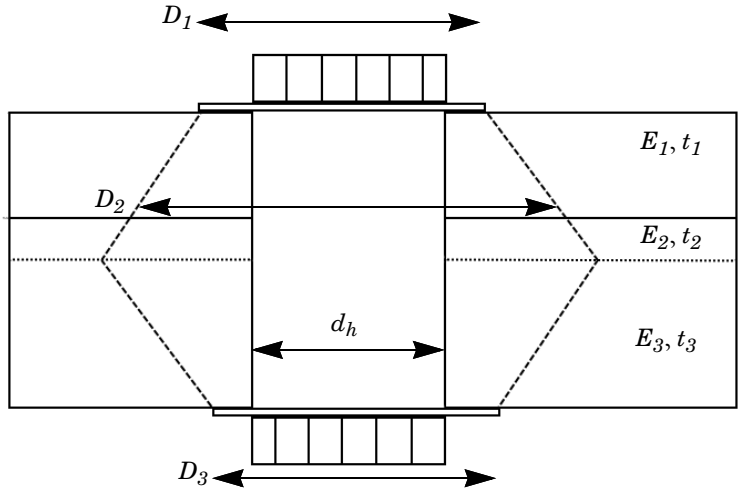


Figure 3-47: Conical pressure profiles for plates of different thicknesses.

For each frustum, the stiffness is defined by

$$k_i = \frac{\pi E_i \tan\alpha_{sh}}{\log\left(\frac{2t_i\tan\alpha_{sh} + D_i - d_h}{2t_i\tan\alpha_{sh} + D_i + d_h} \cdot \frac{D_i + d_h}{D_i - d_h}\right)}$$

with the thicknesses defined as of the thinnest plate, and

$$t_1 = \min(t_{src}, t_{dst})$$

$$t_2 = \frac{|t_{src} - t_{dst}|}{2}$$

$$t_3 = \frac{t_{src} + t_{dst}}{2}$$

and

$$D_1 = D_3 = d_w$$

$$D_2 = d_w + 2\tan\alpha_{sh} t_1$$

The grip stiffness is approximated by giving a value for the frustum angle α_{sh}, which can also depend on the material, the thickness of the plates, and the applied external load.

The displacement difference is defined using the average hole displacement on the two sides:

$$\Delta\mathbf{u} = \tilde{\mathbf{u}}_{\mathrm{dst}} - \tilde{\mathbf{u}}_{\mathrm{src}}$$

Using the normal to the shell on the destination side, $\mathbf{n}$, the displacement difference can be separated into normal and tangential parts:

$$\Delta\mathbf{u}_{\mathrm{n}} = (\Delta\tilde{\mathbf{u}} \cdot \mathbf{n})\mathbf{n}$$

$$\Delta\mathbf{u}_{\mathrm{t}} = \Delta\mathbf{u} - \Delta\mathbf{u}_{\mathrm{n}}\mathbf{n}$$

The contribution to the virtual work for the fastener is then

$$\delta W = \int_A [-k_{\mathrm{n}}\Delta\mathbf{u}_{\mathrm{n}}\mathrm{test}(\Delta\mathbf{u}_{\mathrm{n}}) - k_{\mathrm{s}}\Delta\mathbf{u}_{\mathrm{t}} \cdot \mathrm{test}(\Delta\mathbf{u}_{\mathrm{t}})]dA$$

In presence of a pretension load, the contribution become

$$\delta W = \int_A \left[-\left(\frac{k_{\mathrm{n}} + k_g}{k_{\mathrm{n}}} F_{pre}\mathbf{n}_{dst} + k_{\mathrm{n}}\Delta\mathbf{u}_{\mathrm{n}}\right)\mathrm{test}(\Delta\mathbf{u}_{\mathrm{n}}) + k_{\mathrm{s}}\Delta\mathbf{u}_{\mathrm{t}} \cdot \mathrm{test}(\Delta\mathbf{u}_{\mathrm{t}})\right]dA$$

Automatic Hole Detection

When automatic detection of fastener holes is used in the **Hole Selection** subnode under **Fasteners**, the algorithm attempts to find matching hole pairs on the source and destination boundaries. For two holes to be considered as a pair, and thus connected, the following criteria must be met:

- Both holes are circular.
- They have diameter d within a certain tolerance ε_{d} from the fastener diameter d_{f}. The default limit is $\varepsilon_{\mathrm{d}} = 10\%$.

$$d_f \le d \le d_f(1+\varepsilon_d)$$

- The boundaries are locally parallel. The angle between the two normal vectors must be less than α, with the default value $\alpha = 2°$.

$$\left|\mathbf{e}_{\mathrm{dst}} \times \mathbf{e}_{\mathrm{src}}\right| \le \sin\alpha$$

- The centers of the two holes must lie on a common line. This line is represented by the normal to the source boundary, passing through the center of the hole in the source boundary. What is measured is the ratio between the projection of the distance between the two hole centers, **b**, on that line, and the fastener diameter. The default allowed value is $\varepsilon_c = 1\%$.

$$\left|\mathbf{b} \times \mathbf{e}_{\mathrm{src}}\right| \le \varepsilon_c d_f$$

- For the boundary selection type, an additional filter criterion can be applied to restrict the chosen boundaries to those forming an annular shape around the hole. In this case, the distance between the center of the inner circle, $\mathbf{x}_c$, and the center of the outer circle, $\mathbf{x}_{c,\,a}$, must not exceed a specified fraction of the fastener diameter. The default allowed value is $\varepsilon_c = 1\%$.

$$\left|\mathbf{x}_{c,\,a} - \mathbf{x}_c\right| \le \varepsilon_c d_f$$

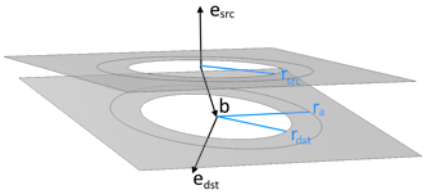


Figure 3-48: Symbols used for hole detection tolerances.

Force Evaluation

For each fastener, axial and shear forces are defined as

$$F_{\mathrm{n}} = -k_{\mathrm{n}}\Delta\mathbf{u}_{\mathrm{n}}$$
$$F_{\mathrm{s}} = -k_{\mathrm{t}}\Delta\mathbf{u}_{\mathrm{t}}$$

When a pretension force is included in the fastener, the normal force becomes

$$F_{\mathrm{n}} = -k_{\mathrm{n}}\Delta\mathbf{u}_{\mathrm{n}} - \frac{k_{\mathrm{n}} + k_{\mathrm{g}}}{k_{\mathrm{n}}} F_{\mathrm{pre}}\mathbf{n}_{\mathrm{dst}}$$

Safety Factors

By adding a **Safety** subnode under **Fasteners**, you can evaluate the degree of utilization for each fastener.

The damage criterion used is

$$\left(\frac{F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} = 1$$

Here $F_{\mathrm{n,max}}$ and $F_{\mathrm{s,max}}$, are the critical forces in tension and shear, respectively. The exponents α_{n} and α_{s} are often chosen as 2 for rivets, while other choices are common for bolts.

For computing a safety factor, s, the following nonlinear equation is solved:

$$\left(\frac{sF_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{sF_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} = 1$$

Stress Linearization

Stress linearization is a procedure in which the stress distribution along a line through the thickness in a solid is approximated with an equivalent linear stress distribution, similar to what would be the result of an analysis using shell theory. The line is commonly referred to as a *stress classification line*, SCL.

First, the computed stresses are transformed into a local orthonormal coordinate system (x_1, x_2, x_3), where the x_1 direction is oriented along the SCL. In 3D, you must specify the x_2 direction, and thus implicitly the x_3 direction. In 2D and 2D axial symmetry, the x_3 direction is in the out-of-plane direction — that is, the Z and azimuthal directions, respectively.

The length of the SCL, which is assumed to be straight, start on one boundary, and end on the opposite boundary, is denoted L.

The *membrane stress tensor* is the average of each local stress component along the SCL:

$$\sigma_{\mathrm{m},ij} = \frac{1}{L}\int_0^L \sigma_{ij} dx_1$$

Each component of the *bending stress tensor* is assumed to have a linear variation along the SCL, with the value being zero at the midpoint.

$$\sigma_{\mathrm{b},ij} = \left(1 - \frac{2x_1}{L}\right)\sigma_{\mathrm{b(max)},ij}$$

The maximum bending stress is defined so that the linear stress distribution has the same moment as the true stress distribution.

$$\sigma_{\mathrm{b(max)},ij} = \frac{6}{L^2}\int_0^L \sigma_{ij}\left(\frac{L}{2} - x_1\right) dx_1$$

The linearized stress distribution is the sum of the membrane and bending stresses,

$$\sigma_{\mathrm{mb},ij}(x_1) = \sigma_{\mathrm{m},ij} + \sigma_{\mathrm{b},ij}(x_1)$$

Finally, the *peak stress tensor* is defined at the two endpoints of the SCL. It is the difference between the actual stress tensor, and the linearized stress representation.

$$\sigma_{\mathrm{p(start)},\,ij} = [\sigma_{ij} - \sigma_{\mathrm{mb},\,ij}]_{x_1 = 0} = [\sigma_{ij}]_{x_1 = 0} - \sigma_{\mathrm{m},\,ij} + \sigma_{\mathrm{b(max)},\,ij}$$
$$\sigma_{\mathrm{p(end)},\,ij} = [\sigma_{ij} - \sigma_{\mathrm{mb},\,ij}]_{x_1 = L} = [\sigma_{ij}]_{x_1 = L} - \sigma_{\mathrm{m},\,ij} - \sigma_{\mathrm{b(max)},\,ij}$$

The *stress intensity*, also known as the Tresca equivalent stress, is often the ultimate goal of a stress linearization. It is computed as a worst case of the equivalent stress at the two ends of the SCL. The stress intensity is computed from the principal stresses as

$$\sigma_{\mathrm{int}} = \sigma_{\mathrm{p1}} - \sigma_{\mathrm{p3}}$$

When computing the principal stresses at the endpoints of the SCL, it is customary to ignore the bending part of the through-thickness oriented stresses. The principal stresses and the stress intensity variables are the endpoints are thus computed using the following stress tensor:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} \sigma_{\mathrm{m},\,11} \\ \sigma_{\mathrm{mb},\,22} \\ \sigma_{\mathrm{mb},\,33} \\ \sigma_{\mathrm{m},\,12} \\ \sigma_{\mathrm{m},\,13} \\ \sigma_{\mathrm{mb},\,23} \end{bmatrix}$$

Section forces, similar to what would be computed in a shell or plate analysis, are another type of result quantities available for each SCL. The in-plane forces are computed from the membrane stresses as

$$\begin{bmatrix} N_{22} \\ N_{33} \\ N_{23} \end{bmatrix} = L \begin{bmatrix} \sigma_{\mathrm{m},\,22} \\ \sigma_{\mathrm{m},\,33} \\ \sigma_{\mathrm{m},\,23} \end{bmatrix}$$

The bending moments are computed from the bending stresses as

$$\begin{bmatrix} M_{22} \\ M_{33} \\ M_{23} \end{bmatrix} = \frac{L^2}{6} \begin{bmatrix} \sigma_{\mathrm{b(max)},\,22} \\ \sigma_{\mathrm{b(max)},\,33} \\ \sigma_{\mathrm{b(max)},\,23} \end{bmatrix}$$

The out-of-plane shear forces are computed from the membrane stresses as

$$\begin{bmatrix} Q_2 \\ Q_3 \end{bmatrix} = L \begin{bmatrix} \sigma_{\mathrm{m},\,12} \\ \sigma_{\mathrm{m},\,13} \end{bmatrix}$$

J-Integral Theory

The J-integral is a path independent integral, used for characterizing the severity of the loading at a crack tip. It can be used both in an elastic and an elastoplastic analysis. In the case of elasticity, the value of the J-integral can be shown to be identical to the strain energy release rate G, and also to be related to the stress intensity factor K.

In 2D, the J-integral is defined as

$$J = \int_{\Gamma_{tot}} \left[W_s dy - (\sigma \cdot \mathbf{m}) \cdot \frac{\partial \mathbf{u}}{\partial x} ds \right] \tag{3-242}$$

where W_s is the strain energy density, σ is a stress tensor, and $\mathbf{m}$ is the outward normal of the integration contour Γ. In Equation 3-242, the crack is assumed to extend in the positive x direction, and integration is made over $\Gamma_{tot} = \Gamma \cup \Gamma_{face}$, which can be any closed path around crack tip, see Figure 3-49. However, if there is no loading on the crack face, the contribution to J on Γ_{face} is zero, and it is sufficient to only perform the integration over Γ.

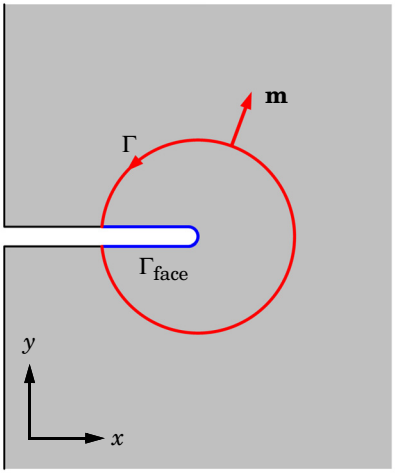


Figure 3-49: Circular J-integral around a notch in two dimensions.

For a general crack extension direction $\mathbf{p}$, Equation 3-242 can be generalized to

$$J = \int_{\Gamma} [(\mathbf{m} \cdot \mathbf{p}) W_s - (\sigma \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] ds$$

In 3D, the path integral can be taken around an arbitrary point of the crack front, and Γ_{tot} can be any closed path around the point in the normal plane to the crack front. It then turns out that there is also an additional term, to be integrated over the area enclosed by Γ_{tot}. The expression is then further extended to

$$J = \int_{\Gamma} [(\mathbf{m} \cdot \mathbf{p}) W_s - (\sigma \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] \, ds - \int_{A} \nabla [(\sigma \cdot \mathbf{t}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] \cdot \mathbf{t} \, dA$$

where $\mathbf{t}$ is the tangent to the crack front. In practice, this is also the normal to the plane of the integration contour.

In 2D axisymmetry, $\mathbf{m}$ and $\mathbf{p}$ are located in the rz-plane, and $\mathbf{t}$ is oriented in the azimuthal direction. In this case, the full expression can be written as

$$J = \int_{\Gamma} [(\mathbf{m} \cdot \mathbf{p}) W_s - (\sigma \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] \, ds - (\mathbf{e}_r \cdot \mathbf{p}) \int_{A} [W_s - \sigma_{\phi\phi} (\nabla \mathbf{u})_{\phi\phi} - \sigma_{r\phi} (\nabla \mathbf{u})_{r\phi}] dA$$

where $\mathbf{e}_r$ is the base vector of the global r direction.

LOADS ON CRACK FACES

When a boundary load $\mathbf{F}_A$ is applied on the crack face, it contributes with an additional term to the J-integral since the contour integrals on the crack face become nonzero. This additional contribution, J_{face}, is defined by

$$J_{face} = - \int_{\Gamma_{face}} \mathbf{F}_A \cdot (\nabla \mathbf{u} \cdot \mathbf{p}) \, ds$$

where Γ_{face} is schematically defined in Figure 3-49.

COMPUTING STRESS INTENSITY FACTORS FROM J-INTEGRAL

From the J-integral, it is possible to compute the stress intensity factor K for the three modes of fracture:

- Crack opening, or mode I, is described by stress intensity factor K_{I}
- Crack sliding, or mode II, is described by stress intensity factor K_{II}
- Tearing, or mode III, is described by stress intensity factor K_{III}

Computation of the stress intensity factors is done by utilizing the following decomposition of the J-integral

$$J = J_{\mathrm{I}} + J_{\mathrm{II}} + J_{\mathrm{III}} = \frac{1}{E_{\mathrm{eff}}}\left(K_{\mathrm{I}}^2 + K_{\mathrm{II}}^2 + \frac{1}{1-\nu}K_{\mathrm{III}}^2\right) \tag{3-243}$$

where E_{eff} is an effective Young's modulus that accounts for the stress state at the crack front. Assuming that $K_{\mathrm{II}} = \beta K_{\mathrm{I}}$ and $K_{\mathrm{III}} = \gamma K_{\mathrm{I}} / (1 + \nu)$, where β and γ are coefficients that account for the mode mixture, it follows from Equation 3-243 that the stress intensity factors are

$$\begin{aligned}
K_{\mathrm{I}} &= \sqrt{\frac{E_{\mathrm{eff}}}{1 + \beta^2 + \frac{1}{1+\nu}\gamma^2}J} \\
K_{\mathrm{II}} &= \sqrt{\frac{E_{\mathrm{eff}}}{1 + \frac{1}{\beta^2} + \frac{1}{1+\nu}\frac{\gamma^2}{\beta^2}}J} \\
K_{\mathrm{III}} &= \sqrt{\frac{E_{\mathrm{eff}}}{(1+\nu)\left[1 + \frac{1+\nu}{\gamma^2} + \frac{(1+\nu)\beta^2}{\gamma^2}\right]}J}
\end{aligned} \tag{3-244}$$

The mode mixture coefficients are defined by

$$\beta = \frac{\Delta u_{\mathrm{II}}}{\Delta u_{\mathrm{I}}}$$

$$\gamma = \frac{\Delta u_{\mathrm{III}}}{\Delta u_{\mathrm{I}}}$$

where Δu_{I}, Δu_{II}, and Δu_{III} are the components of the displacement vector across the crack, $\Delta\mathbf{u}$, defined in the coordinate system corresponding to the three modes of fracture.

In 2D, mode III is not relevant, and Equation 3-244 simplifies to

$$K_{\mathrm{I}} = \sqrt{\frac{E_{\mathrm{eff}}}{1+\beta^2}J}$$

$$K_{\mathrm{II}} = \sqrt{\frac{E_{\mathrm{eff}}}{1+\frac{1}{\beta^2}}J}$$

The definition of the effective Young's modulus depends on the stress state at the crack tip

$$E_{\mathrm{eff}} = \begin{cases} \dfrac{E}{1-\nu^2} & \text{if plane strain} \\ E & \text{if plane stress} \end{cases}$$

Hence, in 2D it depends on the 2D approximation used in the physics interface, while in 3D it is assumed that the stress state is approximately plane strain.

Virtual Crack Extension Theory

The virtual *crack extension method* is a way to compute the energy release rate for a crack. The idea in the virtual crack extension method is to evaluate the change in total energy when the surface area of a crack is changed, but without actually solving for different geometries. Several different variants on this theme can be found in the literature.

The energy release rate, G, is defined as

$$G = -\frac{\partial U}{\partial A}$$

where U is the total potential energy, and A is an increase in crack area. The total potential energy consists of the strain energy and the potential of the loads.

$$U = \int_V W_s \, \mathrm{dV} - \left(\int_V \mathbf{f}_\mathrm{v} \cdot \mathbf{u} \, \mathrm{dV} + \int_S \mathbf{f}_\mathrm{s} \cdot \mathbf{u} \, \mathrm{dS} + \int_L \mathbf{f}_l \cdot \mathbf{u} \, \mathrm{dL} + \sum_i \mathbf{f}_i \cdot \mathbf{u}_i \right)$$

For a 2D case, the interpretation is simple, since the crack length, a, is a scalar. Then,

$$G = -\frac{1}{d}\frac{\partial U}{\partial a}$$

where d is the thickness.

For a crack in 3D, the energy release rate must be viewed as local quantity that varies along the crack front. There is no well defined crack length. Let s be a curve parameter along the crack front. Then

$$G(s) = -\frac{\partial U}{\partial A(s)}$$

Here $A(s)$ is some kind of local crack area. The new area that is created by a small crack growth must be located in the plane of the crack, and by extending the crack perpendicular to the crack front. If $\mathbf{s}$ is the tangent to the crack front, and $\mathbf{n}(s)$ is the local normal to the crack face, then the local crack extension direction is

$$\mathbf{a}(s) = \mathbf{s} \times \mathbf{n}(s)$$

Of course, it is not possible to create a new area of the crack in a pointwise fashion. It must have some, arbitrary, extension along the crack front. In a finite element context, the local crack extension is typically distributed over one or two elements.

In the current implementation, the built-in functionality for sensitivity analysis is used together with a deformed geometry. In a small cylindrical region around the crack tip, the *sensitivity region*, there is a prescribed deformation field. The prescribed deformation, **dX** inside the domain is

$$\mathbf{dX} = c(s)\left(1 - \frac{r}{R_s}\right)\mathbf{a}(s)$$

Here, r is the distance from the crack tip in the plane perpendicular to the crack tip. R_s is the radius of the sensitivity region, which can be given by the user. The control variable field for the sensitivity analysis is denoted $c(s)$.

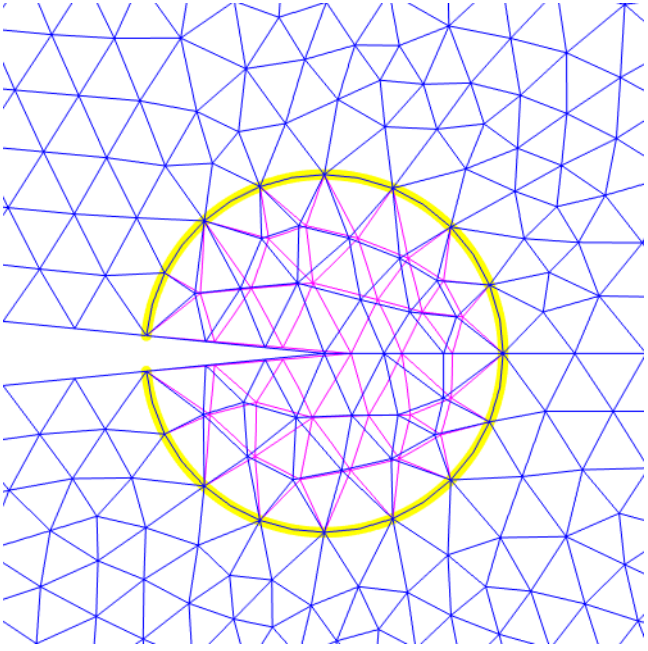


Figure 3-50: The deformed geometry with interpolated mesh. The yellow ring shows the limit of the sensitivity region. The mesh translation is largest at the crack tip and then decreases linearly through the sensitivity region.

The energy release rate is then determined as

$$G = -\frac{\left(\frac{\partial U}{\partial c}\right)}{\left(\frac{\partial A_c}{\partial c}\right)}$$

where A_c is the area of the crack.

The size of the sensitivity region should not be too small. If it just encompasses the elements closest to the crack, the singular fields that cannot be represented well in the solution will influence the sensitivity evaluation too much. On the other hand, it should be relatively small, so that it does not include, for example, exterior boundaries.

COMPUTING STRESS INTENSITY FACTORS FROM ENERGY RELEASE RATE

Stress intensity factors can be computed from the energy release rate in exactly the same way as from a J-integral. The methods described in Computing Stress Intensity Factors From J-Integral are applied also in this case.

Embedded Elements

The Embedded Reinforcement multiphysics coupling provides a generic method to couple lower dimension structural elements to a solid domain. Mesh elements from the Truss, Wire, Beam, or Membrane interfaces within or adjacent to a solid domain can be connected by adding this multiphysics coupling. Two different formulations are available for connecting the interfaces: **Rigid** or **Spring** connection. The rigid connection is based on a pointwise constraint formulation, while the spring connection is based on a penalty regularization of the constraint. When using the spring connection, a nonlinear behavior can be included by adding a bond slip model.

To facilitate the connection between the solid domain and the embedded structure, a nonlocal coupling is set up between the two interfaces. This coupling is made through a **General Extrusion** operator that maps an expression defined on the source to an expression that can be evaluated on the destination geometry. Here, the source is always the solid domain, and the destination is the embedded thin structure. The latter can be either a Truss, Wire, Beam, or Membrane interface. Mathematically, the mapping of a generic expression $\boldsymbol{E}$ from a source domain Ω_{src} to a point $\mathbf{X}_{\text{dst}}$ on the destination boundary (or edge) Γ_{dst} can be described by

$$\boldsymbol{E}_{\text{src}} = \text{map}[\boldsymbol{E}(\mathbf{X}_{\text{src}}), \mathbf{X}_{\text{dst}}] \quad \text{on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}} \tag{3-245}$$

The mapping is such that for each point $\mathbf{X}_{\text{dst}}$, it finds the closest point $\mathbf{X}_{\text{src}}$ in Ω_{src} at which to evaluate $\boldsymbol{E}$. From Equation 3-245 it is inferred that the mapping is made in the material frame, that is, the mapping is constant even for a geometrically nonlinear case. Moreover, the mapping is only valid for the part of Γ_{dst} that lies within Ω_{src}.

With the definition in Equation 3-245, the rigid connection type is described by adding a constraint on Γ_{dst} such that the displacement field $\mathbf{u}$ equals

$$\mathbf{u} = \text{map}[\mathbf{u}(\mathbf{X}_{\text{src}}), \mathbf{X}_{\text{dst}}] \text{ on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

The penalty connection is set up by first defining a relative displacement vector $\Delta\mathbf{u}$ between the mapped source and the destination

$$\Delta\mathbf{u} = \text{map}[\mathbf{u}(\mathbf{X}_{\text{src}}), \mathbf{X}_{\text{dst}}] - \mathbf{u}(\mathbf{X}_{\text{dst}}) \quad \text{on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

From the relative displacement vector $\Delta\mathbf{u}$, the spring force $\mathbf{f}$ of the penalty formulation is defined by Hooke's law

$$\mathbf{f} = K\Delta\mathbf{u} \quad \text{on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

where K is the spring stiffness matrix, which has nonzero components on the diagonal only. The spring stiffness matrix K is most naturally described in the local coordinate system of the thin structure, Γ_{dst}, such that the spring force $\mathbf{f}_\text{l}$ in local coordinates is defined by

$$\mathbf{f}_\text{l} = K_\text{l}(T \cdot \Delta\mathbf{u}) = K_\text{l}\Delta\mathbf{u}_\text{l}$$

where T is the transformation matrix form local to global coordinates and K_l is the spring stiffness matrix in the local coordinate system. The local coordinate system is either defined by the destination physics interface, or automatically by the multiphysics coupling. Finally, the following weak contribution is added

$$0 = \int_{\Gamma_{\text{dst}}} -\mathbf{f} \cdot \delta\Delta\mathbf{u} dS$$

When a **Bond slip model** is added to the spring connection, an additive decomposition of $\Delta\mathbf{u}$ into elastic and plastic counterparts is assumed so that, in the local coordinate system

$$\Delta\mathbf{u}_{\text{l, e}} = \Delta\mathbf{u}_\text{l} - \mathbf{u}_{\text{l, p}} \quad \text{on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

where $\mathbf{u}_{\text{l,e}}$ is the elastic displacement, and $\mathbf{u}_{\text{l,p}}$ is the plastic displacement (or slip) on the destination. The definitions of the spring force and the weak contribution are unaffected.

The slip vector $\mathbf{u}_{\text{l,p}}$ is defined through a local constitutive model based on the flow theory of plasticity. When the destination physics is either a Truss, Wire, or a Beam interface, slip can only occur in the direction of the local edge tangent, which means that the slip vector is defined by a single scalar value, that is, $\mathbf{u}_{\text{l,p}} = \{0,0,u_{\text{pn}}\}$. The constitutive model is then summarized by

$$\begin{gathered} F \equiv |f_\text{n}| - c \le 0 \\ \dot{u}_{\text{pn}} = \lambda \frac{\partial F}{\partial f_\text{n}} \\ \dot{u}_{\text{pe}} = \lambda \\ F \le 0,\ \lambda \ge 0,\ \lambda F = 0 \end{gathered} \tag{3-246}$$

where f_n and u_{pn} is the spring force and slip in the direction of the local edge tangent, respectively. The cohesion c determines the sliding resistance, and it can depend on any variable or field present in the model. The plasticity model is defined by the yield function F, the plastic multiplier λ, and the accumulated slip u_{pe}. The last row of Equation 3-246 are the Kuhn–Tucker conditions.

When the destination physics is a Membrane interface, sliding can only occur in the direction of the local tangents of the plane given by the normal of the boundary, that is, $\mathbf{u}_{l,p} = \{u_{t1}, u_{t2}, 0\}$. The constitutive model then takes a slightly different form, and it can be summarized by

$$\begin{gathered} F \equiv \|\mathbf{f}_t\| - c \le 0 \\ \left\{ \dot{u}_{t1,\,p} \;,\; \dot{u}_{t2,\,p} \;,\; 0 \right\} = \lambda \frac{\partial F}{\partial \mathbf{f}_t} \\ \dot{u}_{pe} = \lambda \\ F \le 0,\; \lambda \ge 0,\; \lambda F = 0 \end{gathered} \tag{3-247}$$

where $\mathbf{f}_t$ is a vector containing the two tangent components of the local spring force $\mathbf{f}_l$.

The rate equations in the constitutive models given by Equation 3-246 and Equation 3-247 are implemented through a backward Euler discretization. The resulting set of nonlinear algebraic equations is solved using Newtons method to find λ, u_{pe}, and $\mathbf{u}_{l,p}$. Optionally, it is also possible to compute the dissipated energy due to friction, W_p, by also solving the following rate equation

$$\dot{W}_p = \lambda \frac{\partial F}{\partial \mathbf{f}_l} \cdot \mathbf{f}_l$$

When the **Embedded structure** is a Beam interface, there is an option to suppress rotation around the beam axis. This option adds a constraint for the rotation θ_{xl} around the beam axis. The constraint is implemented using a penalty regularization, which adds the following weak contribution to the model

$$0 = \int_{\Gamma_{dst}} \frac{GJ}{1000A}\theta_{lx}\delta\theta_{lx}dS$$

where GJ is the torsional stiffness of the beam and A is the area of the beam. This functionality is intended to suppress unwanted rigid body rotations that can lead to nonconvergence of the solver.

Thin Layers

Sometimes, there are thin layers between larger domains that cannot be ignored in a structural analysis. The reasons are either because they have significant flexibility or damping properties, or that they may fail. Examples of such layers are glue layers, O-rings, and gaskets.

When the layer is thin compared to other dimensions of the structure, it is difficult to produce a mesh with acceptable quality and without compromising the numerical conditioning of the stiffness matrix.

The Thin Layer boundary feature allows to overcome this problem, through simplifications in the definition of the Deformation Measures, thus avoiding meshing in the through-thickness direction.

There are three different formulations available for modeling thin layers:

- Solid Approximation
- Membrane Approximation
- Spring Approximation

Deformation Gradient

The deformation gradient $\boldsymbol{F}$ is defined as the gradient of the current coordinates with respect to the original coordinates:

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = I + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$$

See the Lagrangian Formulation section for details.

For thin layers, the deformation gradient is approximated as

$$F = I + \nabla_t \mathbf{u} + (\nabla \mathbf{u} \cdot \mathbf{N}) \otimes \mathbf{N}$$

Here, $\nabla_t \mathbf{u}$ is the displacement gradient computed using the tangential gradient on the reference surface, $\mathbf{N}$ is the normal vector to the undeformed surface, and $\otimes$ is the outer product of two vectors, $(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j$. The term $\nabla_n \mathbf{u} = (\nabla \mathbf{u} \cdot \mathbf{N}) \otimes \mathbf{N}$ is the so-called normal gradient, that can be approximated by different approaches.

Solid Approximation

For the **Solid** approximation on interior thin layers, the displacements between the two sides of the boundary are decoupled. The two boundaries are then connected by elastic and viscous forces proportional to the relative displacements and the corresponding material model.

By using the notation $\mathbf{u}_1 = \texttt{down}(\mathbf{u})$ and $\mathbf{u}_2 = \texttt{up}(\mathbf{u})$, so $\mathbf{u}_1$ represents the displacement of the downside (bottom), and $\mathbf{u}_2$ is the displacement of the upside (top) of the thin layer, the average displacement of the midsurface reads

$$\mathbf{u}_\mathrm{a} = \frac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2)$$

and the extension of the thin layer reads

$$\mathbf{u}_\mathrm{e} = \mathbf{u}_2 - \mathbf{u}_1$$

With this notation, the normal vector $\mathbf{N}$ points from the downside (bottom) toward the upside (top).

The gradient through the normal direction is approximated as

$$\nabla_\mathrm{n}\mathbf{u} = (\nabla\mathbf{u}\cdot\mathbf{N})\otimes\mathbf{N} \approx \frac{1}{d}\mathbf{u}_\mathrm{e}\otimes\mathbf{N}$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{a}\otimes\mathbf{b})_{ij} = a_i b_j$, d is the thickness of the thin layer, and the quantity $\mathbf{u}_\mathrm{e}/d$ represents its strain.

The deformation gradient in the layer is then approximated as

$$F = I + \nabla_\mathrm{t}\mathbf{u}_\mathrm{a} + \frac{1}{d}\mathbf{u}_\mathrm{e}\otimes\mathbf{N}$$

The tangential deformation gradient on the midsurface is calculated as

$$F_\mathrm{t} = I - \mathbf{N}\otimes\mathbf{N} + \nabla_\mathrm{t}\mathbf{u}_\mathrm{a}$$

and the normal deformation gradient as

$$F_\mathrm{n} = \mathbf{N}\otimes\mathbf{N} + \frac{1}{d}\mathbf{u}_\mathrm{e}\otimes\mathbf{N}$$

The normal deformation gradient contains both normal as well as transverse shear deformations, and it is constant through the thickness of the layer. The tangential

deformation gradient, F_t, contains information about the stretching in the midsurface, the so-called membrane deformation.

When **Layered** is selected in the **Thin layer type** list, the gradient through the normal direction is approximated as

$$\nabla_n \mathbf{u} = (\nabla \mathbf{u} \cdot \mathbf{N}) \otimes \mathbf{N} \approx \frac{\partial \mathbf{u}}{\partial z} \otimes \mathbf{N}$$

where z is the extra dimension coordinate, that varies between 0 and the layer thickness d.

The deformation gradient in the layered thin structure is then approximated as

$$F = I + \nabla_t \mathbf{u} + \frac{\partial \mathbf{u}}{\partial z} \otimes \mathbf{N}$$

The weak contributions are scaled by a correcting factor that accounts for the curvature of the thin layer.

The layered formulation can handle composites and laminates, see the Theory for the Layered Shell Interface section for details.

Membrane Approximation

For the **Membrane** approximation, there is no slit between the two sides of the thin layer. The tangential deformation gradient is computed on the surface as

$$F_t = I - \mathbf{N} \otimes \mathbf{N} + \nabla_t \mathbf{u}$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{N} \otimes \mathbf{N})_{ij} = N_i N_j$.

The tangential deformation gradient is augmented to include the normal stretch

$$F_n = \lambda_n \mathbf{n} \otimes \mathbf{N}$$

and the transverse shear deformation

$$F_s = (\gamma_1 \mathbf{t}_1 + \gamma_2 \mathbf{t}_2) \otimes \mathbf{N}$$

Here, λ_n is the transverse (normal) stretch, and γ_1 and γ_2 are the transverse shears. The vectors $\mathbf{n}$, $\mathbf{t}_1$, and $\mathbf{t}_2$ are the normal and tangent unit vectors on the deformed surface.

See the Theory for the Membrane Interface section for details.

Spring Approximation

For the **Spring** approximation, the membrane deformation is neglected, and the deformation gradient reads

$$F = I + \frac{1}{d}\mathbf{u}_c \otimes \mathbf{N}$$

where $\otimes$ is the outer product of two vectors, $(\mathbf{a}\otimes\mathbf{b})_{ij} = a_i b_j$. Here, the thin layer extension $\mathbf{u}_e$ contains information about the normal as well as the transverse shear deformation, but no membrane deformation is taken into account.

In case of predeformation, the extension of the thin layer reads

$$\mathbf{u}_c = \mathbf{u}_2 - \mathbf{u}_1 - \mathbf{u}_0$$

where $\mathbf{u}_0$ is the predeformation of the thin layer.

Material Models in Thin Layers

After selecting the approximation, **Solid**, **Membrane**, or **Spring**, the strain tensor in a geometrically linear analysis is computed from

$$\varepsilon = \frac{1}{2}(F^T + F) - I$$

and for a geometrically nonlinear analysis from

$$\varepsilon = \frac{1}{2}(F^T F - I)$$

Then the material models are defined by the corresponding strain tensor.

There are some cases when a small strain formulation could be useful, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient to describe the thin layer.

See the Linear Elastic Material, Nonlinear Elastic Materials, and Hyperelastic Materials sections for details.

For anisotropic data, the coefficients are given with respect to the boundary system. See the Boundary System section for details.

You can incorporate inelastic effects by adding the following subnodes to a **Linear Elastic Material** or **Nonlinear Elastic Material** node in thin layers:

- Thermal Expansion and Thermoelastic Damping
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Linear Viscoelasticity
- Elastoplastic Materials
- Fibers for Linear and Nonlinear Elastic Materials
- Creep
- Viscoplasticity
- Damage Models (Linear Elastic Materials)
- Safety Factor Evaluation

The following subnodes are available for a **Hyperelastic Material**:

- Thermal Expansion and Thermoelastic Damping
- Hygroscopic Swelling
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Large Strain Viscoelasticity
- Elastoplastic Materials
- Creep and Viscoplasticity for Large Strains
- Polymer Viscoplasticity
- Fibers for Hyperelastic Materials
- Mullins Effect

Spring Material

When the **Spring** approximation is chosen, it is possible to specify a *Spring material* for the thin layer.

In the geometrically linear case, the spring force $\mathbf{f}_s$ per reference area is computed as

$$\mathbf{f}_s = -K_A \mathbf{u}_c$$

where K_A is the spring constant. The spring constant can be a scalar or a matrix which components are the spring constants given with respect to the boundary system. When using the boundary system to specify the spring constants, this matrix reads.

$$K_{\mathrm{A}} = \begin{bmatrix} k_{\mathrm{t1}} & k_{\mathrm{t1t2}} & k_{\mathrm{t1n}} \\ k_{\mathrm{t1t2}} & k_{\mathrm{t2}} & k_{\mathrm{t2n}} \\ k_{\mathrm{t1n}} & k_{\mathrm{t2n}} & k_{\mathrm{n}} \end{bmatrix}$$

See the Boundary System section for details.

For the geometric nonlinear case, the spring force reads

$$\mathbf{f}_s = -K_{\mathrm{A}}\left(\mathbf{u}_{\mathrm{c}} + \frac{|\mathbf{u}_{\mathrm{c}}|^2}{2d}\mathbf{N}\right)$$

LOSS FACTOR DAMPING

When the **Spring** approximation is chosen, it is possible to add **Loss factor damping** or **Viscous damping** to the spring material.

When using loss factor damping, a complex spring matrix is used to define the spring force. With an isotropic loss factor η_{s}, this means that

$$K^{\mathrm{c}} = (1 + j\eta_{\mathrm{s}})K_{\mathrm{A}}$$

where K_{A} is the constitutive matrix computed from the spring constants, and K^{c} is the complex constitutive matrix used when computing the spring force.

For a linear elastic material, this would be equivalent to multiplying Young's modulus by the factor $(1 + j\eta_{\mathrm{s}})$.

It is also possible to give individual loss factors for each entry in the constitutive spring matrix, so that

$$K^{\mathrm{c}}_{mn} = (1 + j\eta_{\mathrm{s},\,mn})K_{mn}$$

Loss factor damping is only applicable in frequency domain.

VISCOUS DAMPING

Viscous damping causes a viscous stress proportional to the rate of strain in the material,

$$s_{\mathrm{q}} = \eta_{\mathrm{b}}\dot{\varepsilon}_{\mathrm{vol}} + 2\eta_{\mathrm{v}}\dot{\varepsilon}_{\mathrm{dev}}$$

where η_{b} and η_{v} are the normal and shear viscosity coefficients, respectively.

For the **Spring** approximation, only normal extension causes volume changes in the thin layer. The volumetric (normal) strain in the geometric linear case is computed from

$$\varepsilon_{\text{vol}} = \left(\frac{1}{d}\mathbf{u}_{\text{e}} \cdot \mathbf{N}\right)\mathbf{N} \otimes \mathbf{N}$$

and for the geometric nonlinear case from

$$\varepsilon_{\text{vol}} = \left(\frac{1}{d}\mathbf{u}_{\text{e}} \cdot \mathbf{N} + \frac{|\mathbf{u}_{\text{e}}|^2}{2d^2}\right)\mathbf{N} \otimes \mathbf{N}$$

In both cases, the deviatoric (transverse shear) strain reads

$$\varepsilon_{\text{dev}} = \frac{1}{2d}\mathbf{u}_{\text{e}} \otimes \mathbf{N} + \frac{1}{2d}\mathbf{N} \otimes \mathbf{u}_{\text{e}} - \left(\frac{1}{d}\mathbf{u}_{\text{e}} \cdot \mathbf{N}\right)\mathbf{N} \otimes \mathbf{N}$$

The viscous force is then computed for the geometric linear case from

$$\mathbf{f}_{\text{q}} = -\eta_{\text{b}}\frac{1}{d}(\dot{\mathbf{u}}_{\text{e}} \cdot \mathbf{N})\mathbf{N} - \eta_{\text{v}}\frac{1}{d}(\dot{\mathbf{u}}_{\text{e}} - (\dot{\mathbf{u}}_{\text{e}} \cdot \mathbf{N})\mathbf{N})$$

and for the geometric nonlinear case from

$$\mathbf{f}_{\text{q}} = -\eta_{\text{b}}\frac{1}{d}\left(\dot{\mathbf{u}}_{\text{e}} \cdot \mathbf{N} + \frac{1}{d}\mathbf{u}_{\text{e}} \cdot \dot{\mathbf{u}}_{\text{e}}\right)\mathbf{N} - \eta_{\text{v}}\frac{1}{d}(\dot{\mathbf{u}}_{\text{e}} - (\dot{\mathbf{u}}_{\text{e}} \cdot \mathbf{N})\mathbf{N})$$

Viscous damping can be used in both frequency and time domain analyses.

Enclosed Fluids

Fluid loads are commonly encountered in structural analyses. When a closed cavity contains a fluid and the fluid velocity is not as important as the pressure load on the interior walls, the analysis can be simplified as compared to a full Fluid–Structure Interaction analysis. In this case, the fluid volume does not need a mesh.

The fluid load is modeled with a certain assumption about the fluid's compressibility. For instance, it can be modeled as a compressible ideal gas, or as fully incompressible fluid. An extra equation of state is imposed with a weak constraint on the pressure variable.

This technique is useful for modeling rubber parts filled with air or other gases (seals, airbags, balloons), and also for pressure vessels and other fluid containers.

An enclosed cavity is defined by selecting boundaries, which must form a single, closed volume. A cavity may contain disconnected interior solids (which themselves may enclose cavities) as illustrated in the figure below. One **Enclosed Cavity** node can only represent a single cavity. If a models contains multiple cavities, add one **Enclosed Cavity** node for each individual volume.

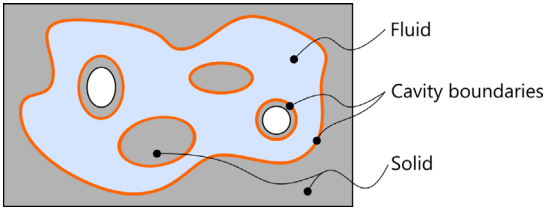


Figure 3-51: A solid (gray) enclosing a cavity filled with a fluid (blue). All solid boundaries adjacent to the cavity (orange) define the volume of the cavity.

Here, a solid encloses a fluid, and additional solid particles are suspended inside the fluid, two of which enclose interior volumes. The primary cavity volume shown in blue, is defined by selecting all solid boundaries adjacent to the cavity. These boundaries, which include the boundaries adjacent to the solid particle boundaries, are highlighted in orange.

The enclosed volume is computed by defining a *flux vector*, which flows through all cavity surfaces. This flux is based on the position vector of the solid in its deformed and undeformed configuration. Using the divergence theorem the volume can be obtained from a surface integral.

Volume Definition

Using the divergence theorem (or Gauss's theorem), the undeformed volume, V_{ref}, of a closed cavity can be computed from only a surface integral rather than a volume integral.

$$V_{\text{ref}} = \int_V d\bar{V} = \int_V (\nabla \cdot \mathbf{R}) d\bar{V} = -\int_{S(V)} (\mathbf{R} \cdot \mathbf{n}) d\bar{S}$$

Here, $S(V)$ is the surface enclosing the cavity, $\mathbf{n}$ is the boundary normal pointing from the solid boundary into the cavity volume; and $\mathbf{R}$ is a suitable vector for which the divergence is equal to 1 for any point on the boundary. For a 3D problem a suitable vector is

$$\mathbf{R} = \frac{1}{3}(\mathbf{X} - \mathbf{X}_{\text{ref}}) = \frac{1}{3}\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} X_{\text{ref}} \\ Y_{\text{ref}} \\ Z_{\text{ref}} \end{bmatrix} \right) \text{ so } \nabla \cdot \mathbf{R} = 1$$

The coordinates of the reference point, $\mathbf{X}_{\text{ref}}$, are constants, which can be useful when dealing with symmetry planes or plane end caps in the model as described below.

The volume of a deformed cavity, V, is obtained in a similar way. By using Nanson's relation the volume can be computed from an integration on the boundaries in the undeformed configuration (material frame).

$$V = \int_v d\bar{v} = \int_v (\nabla_x \cdot \mathbf{r}) d\bar{v} = -\int_{S(V)} (\mathbf{r} \cdot JF^{-T} \cdot \mathbf{n}) d\bar{S}$$

Here, $\boldsymbol{F}$ is the deformation gradient, and $J = \det(\boldsymbol{F})$. The spatial divergence of $\mathbf{r}$ is again constant and equal to 1 for any point on the cavity wall. For 3D $\mathbf{r}$ is defined as

$$\mathbf{r} = \frac{1}{3}(\mathbf{x} - \mathbf{x}_{\text{ref}}) = \frac{1}{3}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_{\text{ref}} \\ y_{\text{ref}} \\ z_{\text{ref}} \end{bmatrix} \right)$$

The integrals shown above are valid for closed surface integrals, that is, when the boundary selection over which to integrate forms a watertight volume. However, often closed volumes may have symmetry planes, or a rigid plane end cap, which is not explicitly modeled. In such cases the true volume is closed, while the model geometry

is no longer watertight. The relations shown above do still apply in such cases, with some minor modifications.

When the model geometry does not form a closed surface, the vectors **R** and **r** must be chosen such that they are parallel along any symmetry plane or plane end cap. This can be achieved by shifting these vectors by selecting an appropriate reference point, $\mathbf{X}_{\text{ref}}$. In addition, the computed volume must be multiplied by a factor, f_V, to get the true cavity volume. For instance, for half symmetry, the true volume will be twice as large as computed by the surface integral, and thus should be multiplied by a factor of two.

More details on how to handle symmetry planes and end caps is described in Symmetry Planes.

Symmetry Planes

The model geometry may contain mirror and sector symmetries, as well as plane, rigid end caps. In such cases, the model geometry does not form a closed surface and further information is required to compute the total volume. The figure below shows a few such examples in 2D. The first two examples show circular cavities, where only part of the volumes are modeled due to mirror symmetries. The third example shows part of a toroidal structure, where the model is reduced due to sector symmetry. The last case shows a pressure vessel mounted on a rigid foundation, which is not explicitly modeled.

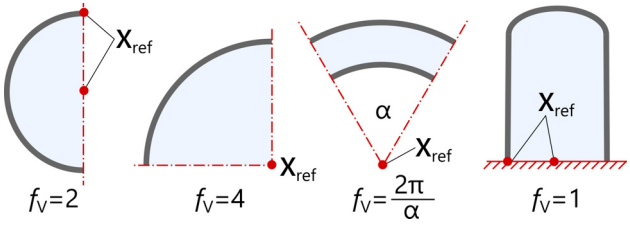


Figure 3-52: Example positions of the reference point and volume factor when modeling cavities bounded by symmetry planes, or rigid end caps/foundation planes.

To compute the volume in the cases shown above, it is necessary to select a reference point, $\mathbf{X}_{\text{ref}}$. The reason is that the flux vector used to compute the volume must not

penetrate any symmetry plane or plane end cap. The following rules apply when selecting the reference point, $\mathbf{X}_{\text{ref}}$:

- For half symmetry (one plane), $\mathbf{X}_{\text{ref}}$ may be an arbitrary point on the symmetry plane. In case the model is 2D axisymmetric, $\mathbf{X}_{\text{ref}}$ is the intersection of the symmetry plane with the axisymmetry axis (that is, $\mathbf{X}_{\text{ref}}$ is located at $R = 0$).
- For quarter symmetry (two orthogonal planes), $\mathbf{X}_{\text{ref}}$ may be an arbitrary point on the intersection of the symmetry planes. In 2D this is a well-defined point.
- For eighth symmetry (three orthogonal planes), $\mathbf{X}_{\text{ref}}$ must be the intersection point of the symmetry planes. This case is only applicable in 3D.
- For rotational symmetry (sector symmetry), $\mathbf{X}_{\text{ref}}$ must be an arbitrary point on the rotational axis. In 2D this is a well-defined point.
- For a plane end cap, $\mathbf{X}_{\text{ref}}$ may be an arbitrary point on the end cap plane. The cavity must only have a single end cap plane.

The volume factor, f_V, shown in the figure above, is defined as the ratio between the true volume and the actual volume included in the model. For example, for a simple mirror symmetry, f_V is equal to two, meaning that the true volume is twice as large as the volume represented by the model geometry.

Equation of State

The volume change between an undeformed and deformed configuration induces a pressure change described by an equation of state. Gases and liquids typically exhibit different compressibility, and are thus modeled differently. Common compressibility models are built-in into the Enclosed Cavity node, and are described below.

COMPRESSIBLE GAS, ADIABATIC

For *adiabatic* expansion or compression of an ideal gas, the deformed volume of the cavity V is related to its initial volume V_{ref} by the equation of state

$$pV^{\gamma} = p_{\text{ref}}V_{\text{ref}}^{\gamma}$$

Here, p_{ref} is the reference fluid pressure inside the undeformed cavity, p is the absolute pressure after deformation, and γ is the *ratio of specific heats*. Since $p = p_{\text{ref}} + p_{\text{rel}}$, this relation is equivalently written as

$$p_{\text{rel}} = p_{\text{ref}}\left[\left(\frac{V_{\text{ref}}}{V}\right)^{\gamma} - 1\right]$$

where p_{rel} is the relative pressure acting on all cavity walls.

COMPRESSIBLE GAS, ISOTHERMAL

For *isothermal* expansion or compression of an ideal gas, the pressure and volume of the deformed cavity are related to the reference pressure and volume by *Boyle's law*:

$$pV = p_{\text{ref}}V_{\text{ref}}$$

This relation is equivalently written as

$$p_{\text{rel}} = p_{\text{ref}}\left(\frac{V_{\text{ref}}}{V} - 1\right)$$

COMPRESSIBLE FLUID

For a compressible fluid, the pressure increment (or decrement) is given by the compressibility of the fluid

$$\chi_{\text{f}} = -\frac{1}{V}\frac{dV}{dp}$$

The relative pressure change is then

$$p_{\text{rel}} = -\frac{1}{\chi_{\text{f}}}\frac{\Delta V}{V} \text{ where } \Delta V = V - V_{\text{ref}}$$

For a nearly incompressible fluid, the fluid compressibility χ_{f} is considered much smaller than the compressibility of the surrounding solid, for instance, a thousand times smaller than a representative bulk modulus.

INCOMPRESSIBLE FLUID

For an incompressible fluid, the pressure variable plays the role of a Lagrange multiplier to the incompressibility constraint

$$V = V_{\text{ref}}$$

Fluid Load

The true force on a surface element acts with a magnitude of the fluid pressure, p, in the normal direction, $\mathbf{n}$, with respect to the current area da:

$$\mathbf{f} = -p\mathbf{n}da$$

The minus sign is applied since the normal vector points from the solid toward the fluid. See Distributed Loads and Fluid–Structure Interaction for details.

Reduced Integration and Hourglass Stabilization

COMSOL Multiphysics uses by default a full and consistent quadrature order when numerically evaluating integrals such as the virtual work. However, a full quadrature order does not necessarily lead to an optimal formulation of the resulting finite element matrices in structural mechanics, where higher-order terms can introduce locking and other detrimental effects. A remedy can be to reduce the quadrature order for selected parts of the virtual work, typically the strain energy density. When full integration is used, the quadrature order is twice the shape order of the displacement field. And when a reduced integration scheme is used, the quadrature order is reduced by half.

Using a reduced integration scheme not only affects the numerical integration of the weak contribution related to the displacement field, but for consistency it also changes the shape order of auxiliary dependent variables such as the out-of-plane strain in plane stress, the auxiliary pressure in a mixed formulation, and other auxiliary dependent variables added by nonlinear material models. This also applies to state variables used in material models such as plasticity, damage, and viscoelasticity. Lowering the order of auxiliary dependent variables and state variables significantly reduces the computation time due to the reduced assembly time.

Using reduced integration affects the stationary part of the virtual work; a full quadrature order is always applied to inertial term contributions in implicit solvers, and mass lumping techniques can be used for explicit dynamics analysis.

Furthermore, reduced integration is applied to weak contributions added by material models. Other contributions to the virtual work such as body loads and other boundary conditions use a full quadrature scheme.

HOURGLASS STABILIZATION

The hourglass stabilization method adds a stabilization term W_{stb} to the strain energy density and to the principle of virtual work.

The exact formulation of W_{stb} differs from case to case, but it is implemented so that it is inexpensive to evaluate. For example, the expression does not include evaluation of nonlinear materials such as plasticity.

ENERGY SAMPLING METHOD

When using reduced integration along with the energy sampling method, the stationary part of the virtual work is written as

$$\delta W = \int_V (-\delta\varepsilon : \sigma) dv + \int_V \delta W_{\text{stb}} dv + \int_V -\delta W_{\text{stb}} dv$$

The additional terms that include W_{stb} are integrated using different quadrature orders. The first and second integrals are evaluated using reduced integration, while the third integral is evaluated using full integration. W_{stb} is chosen such that the combined contribution of the second and third integrals evaluates to zero every time the reduced integration in the first integral is accurate, while it adds a non zero contribution in presence of hourglass modes.

FLANAGAN–BELYTSCHKO METHOD

The Flanagan–Belytschko method (Ref. 53) introduces an artificial stiffness to resist hourglass modes in quadrilateral and hexahedral elements. Hourglass modes are represented as a linear combination of nodal displacement basis.

The method is efficient for controlling hourglass modes without significantly over-stiffening the elements.

HESSIAN METHOD

The strain energy contribution W_{stb} is derived from the norm of the second gradient of the displacement field, also called the *Hessian* of the field.

See also Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

Average Rotation

The **Average Rotation** node is used for computing average rotations and displacements of a virtual rigid body, which is represented by a number of points in the structure. The corresponding velocities and accelerations are also computed.

In the general case, the number of points can be larger than what is needed to define a rigid body movement (3 in 3D and 2 in 2D). Unless the points are attached to a rigid object, it is only possible to determine the rotation in an average sense.

If a certain reference point has a given translation $\mathbf{u}_\mathrm{p}$ and a rotation Θ_p, then the displacement of any other point i can under rigid body assumptions be computed as

$$\tilde{\mathbf{u}}_i = \mathbf{u}_\mathrm{p} + \Theta_\mathrm{p} \times \mathbf{r}_i$$

where $\mathbf{r}_i$ is the distance vector from the reference point to point i. The equation in this form is valid under the assumption of small rotations. In case of large rotations, the displacement due to rotation is instead described by a rotation matrix R:

$$\tilde{\mathbf{u}}_i = \mathbf{u}_\mathrm{p} + (R(\Theta_\mathrm{p}) - I)\mathbf{r}_i$$

Here, I is the identity matrix. The rotation matrix is described with a *quaternion* formulation where parameters a, b, c, and d are components of a unit quaternion.

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix}$$

An extra constraint must be added to enforce a unit quaternion.

$$a^2 + b^2 + c^2 + d^2 = 1$$

In 2D, points can only rotate about the z-axis. The rotation matrix requires only one dependent variable, the rotation angle ϕ:

$$R = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The average rotation and displacement for the set of points is determined using the method of least squares. The square of the difference between the actual displacements and the assumed rigid-body displacement is minimized,

$$r^2 = \sum_i (\mathbf{u}_i - \tilde{\mathbf{u}}_i) \cdot (\mathbf{u}_i - \tilde{\mathbf{u}}_i)$$

Here, the $\mathbf{u}_p$ and Θ_p are replaced with the sought-after average displacement $\mathbf{u}_{avg}$ and average rotation Θ_{avg} respectively. The vector Θ_{avg} is the axis-angle representation of the rotation. The magnitude, $\|\Theta_{avg}\|$, is the rotation angle.

The average velocity and acceleration of the set of points are given as the first and second derivative with respect to time.

$$\frac{\partial}{\partial t}\mathbf{u}_{avg} = \dot{\mathbf{u}}_{avg}\,;\ \frac{\partial}{\partial t}\dot{\mathbf{u}}_{avg} = \ddot{\mathbf{u}}_{avg}$$

For infinitesimal rotations, the average angular velocity and acceleration are defined as

$$\frac{\partial}{\partial t}\Theta_{avg} = \dot{\Theta}_{avg}\,;\ \frac{\partial}{\partial t}\dot{\Theta}_{avg} = \ddot{\Theta}_{avg}$$

For large rotations, the average angular velocity vector is defined from the skew-symmetric matrix W

$$W = R^T\dot{R}\,;\ \frac{\partial}{\partial t}\Theta_{avg} = \mathrm{axl}(W)$$

where R is the rotation matrix.

Warpage

The primary goal of warpage evaluation is to measure the deviation from planarity after deformation of a flat surface. However, this feature can also be applied to surfaces of arbitrary shapes.

- First a rigid body rotation of the surface is computed as a best fit in a least squares sense.
- From that rotated average surface, the deformation in the normal direction is computed.

The original (material frame) coordinates of the surface, which can consist of several boundaries, are given as $\mathbf{X}$. After deformation, corresponding points are located at $\mathbf{x}$, where $\mathbf{x} = \mathbf{X} + \mathbf{u}$.

The total displacement $\mathbf{u}$ is decomposed into an average displacement and a warping displacement, so that

$$\mathbf{u} = \mathbf{u}_a + \mathbf{u}_w$$

Finding the Average Rotation

GEOMETRICALLY LINEAR CASE

If an arbitrary reference point 'p' has a certain translation $\mathbf{u}_p$ and rotation θ_p, then the average displacement of any other point can, under the assumption of rigid body motion, be written as

$$\mathbf{u}_a = \mathbf{u}_p + \theta_p \times (\mathbf{X} - \mathbf{X}_p) = \mathbf{u}_p + \theta_p \times \mathbf{r}$$

The rigid body motion parameters $\mathbf{u}_p$ and θ_p can be determined using a least squares fit. The quadratic error is

$$R^2 = \int_S (\mathbf{u} - \mathbf{u}_a) \cdot (\mathbf{u} - \mathbf{u}_a)\, dA = \int_S \mathbf{u} \cdot \mathbf{u}\, dA - 2\int_S \mathbf{u} \cdot \mathbf{u}_a\, dA + \int_S \mathbf{u}_a \cdot \mathbf{u}_a\, dA \tag{3-248}$$

It is possible to solve the least squares problem analytically by minimizing R^2. In that process, the first of the three integrals above is unimportant, since it does not involve the unknowns. The second integral can be simplified to

$$\int_S \mathbf{u} \cdot \mathbf{u}_a \, dA = \int_S \mathbf{u} \cdot (\mathbf{u}_p + \theta_p \times \mathbf{r}) \, dA =$$
$$\mathbf{u}_p \cdot \int_S \mathbf{u} \, dA + \theta_p \cdot \int_S \mathbf{r} \times \mathbf{u} \, dA = \mathbf{u}_p \cdot \mathbf{i}_u + \theta_p \cdot \mathbf{i}_{ru}$$

Similarly, the third integral can be expanded to

$$\int_S \mathbf{u}_a \cdot \mathbf{u}_a \, dA = \int_S (\mathbf{u}_p + \theta_p \times \mathbf{r}) \cdot (\mathbf{u}_p + \theta_p \times \mathbf{r}) \, dA =$$
$$\int_S \mathbf{u}_p \cdot \mathbf{u}_p \, dA + 2\int_S \mathbf{u}_p \cdot (\mathbf{u}_p + \theta_p \times \mathbf{r}) \, dA + \int_S (\theta_p \times \mathbf{r}) \cdot (\theta_p \times \mathbf{r}) \, dA =$$
$$\mathbf{u}_p \cdot \mathbf{u}_p \int_S dA + 2(\mathbf{u}_p \times \theta_p) \cdot \int_S \mathbf{r} dA + (\theta_p \cdot \theta_p) \int_S \mathbf{r} \cdot \mathbf{r} \, dA - \int_S (\theta_p \cdot \mathbf{r})^2 \, dA =$$
$$\mathbf{u}_p \cdot \mathbf{u}_p i_1 + 2(\mathbf{u}_p \times \theta_p) \cdot \mathbf{i}_r + (\theta_p \cdot \theta_p) i_{rr} - \sum_{i,j} \theta_{p,i} \theta_{p,j} \int_S r_i r_j \, dA =$$
$$\mathbf{u}_p \cdot \mathbf{u}_p i_1 + 2(\mathbf{u}_p \times \theta_p) \cdot \mathbf{i}_r + (\theta_p \cdot \theta_p) i_{rr} - \sum_{i,j} \theta_{p,i} \theta_{p,j} i_{ij}$$

To minimize this function, it should be differentiated with respect to all unknowns, and each equation set equal to zero. The first equation is obtained from

$$\frac{\partial R^2}{\partial u_{p,x}} = -i_{u,x} + u_{p,x} i_1 + \theta_{p,y} i_{r,z} - \theta_{p,z} i_{r,y} = 0$$

The next two equations can be obtained by using index permutation. The fourth equation reads

$$\frac{\partial R^2}{\partial \theta_{p,x}} = -i_{ru,x} + u_{p,z} i_{r,y} - u_{p,y} i_{r,z} + (i_{rr} - i_{xx}) \theta_{p,x} - \theta_{p,y} i_{xy} - \theta_{p,z} i_{xz} = 0$$

Finally, the fifth and sixth equations are obtained by using index permutations. This gives the following system of equations on matrix form:

$$\begin{bmatrix} i_1 & 0 & 0 & 0 & i_{\mathrm{r},z} & -i_{\mathrm{r},y} \\ 0 & i_1 & 0 & -i_{\mathrm{r},z} & 0 & i_{\mathrm{r},x} \\ 0 & 0 & i_1 & i_{\mathrm{r},y} & -i_{\mathrm{r},x} & 0 \\ 0 & -i_{\mathrm{r},z} & i_{\mathrm{r},y} & i_{yy}+i_{zz} & -i_{xy} & -i_{xz} \\ i_{\mathrm{r},z} & 0 & -i_{\mathrm{r},x} & -i_{xy} & i_{xx}+i_{zz} & -i_{yz} \\ -i_{\mathrm{r},y} & i_{\mathrm{r},x} & 0 & -i_{xz} & -i_{yz} & i_{xx}+i_{yy} \end{bmatrix} \begin{bmatrix} u_{\mathrm{p},x} \\ u_{\mathrm{p},y} \\ u_{\mathrm{p},z} \\ \theta_{\mathrm{p},x} \\ \theta_{\mathrm{p},y} \\ \theta_{\mathrm{p},z} \end{bmatrix} = \begin{bmatrix} i_{\mathrm{u},x} \\ i_{\mathrm{u},y} \\ i_{\mathrm{u},z} \\ i_{\mathrm{ru},x} \\ i_{\mathrm{ru},y} \\ i_{\mathrm{ru},z} \end{bmatrix}$$

This system of equations can now be directly solved.

GEOMETRICALLY NONLINEAR CASE

In a geometrically nonlinear setting, the cross product must be replaced by a general rotation matrix, $\mathbf{R}$. The rotation matrix is now a nonlinear function of the rotation vector, θ_p. However, another parameterization of the rotation using a quaternion representation is chosen in this case. For more information about the quaternion representation, see the theory for Rigid Connector. The average displacement field can now be written as

$$\mathbf{u}_\mathrm{a} = \mathbf{u}_\mathrm{p} + (\mathbf{R}(\mathbf{q}_\mathrm{p}) - \mathbf{I})(\mathbf{X} - \mathbf{X}_\mathrm{p})$$

The rotation matrix is a function of the four quaternion degrees of freedom, symbolically collected in the vector $\mathbf{q}_\mathrm{p}$.

The quadratic error is, as before

$$R^2 = \int_S (\mathbf{u} - \mathbf{u}_\mathrm{a}) \cdot (\mathbf{u} - \mathbf{u}_\mathrm{a})\, dA \tag{3-249}$$

Using a weak form of the least squares problem,

$$\int_S [\mathbf{u} - (\mathbf{u}_\mathrm{p} + (\mathbf{R}(\mathbf{q}_\mathrm{p}) - \mathbf{I})(\mathbf{X} - \mathbf{X}_\mathrm{p}))] \cdot \delta(\mathbf{u}_\mathrm{p})\, dA = 0$$

$$\int_S [\mathbf{u} - (\mathbf{u}_\mathrm{p} + (\mathbf{R}(\mathbf{q}_\mathrm{p}) - \mathbf{I})(\mathbf{X} - \mathbf{X}_\mathrm{p}))] \cdot \delta(\mathbf{R}(\mathbf{q}_\mathrm{p})(\mathbf{X} - \mathbf{X}_\mathrm{p}))\, dA = 0$$

This constitutes a set of nonlinear equations for determining the average rigid body motion in terms of $\mathbf{u}_\mathrm{p}$ and $\mathbf{q}_\mathrm{p}$. The Jacobian of this system will, in general, cause a full coupling to the displacement field $\mathbf{u}$ and it is also nonsymmetric. Since the structural mechanics problem in itself often has a symmetric stiffness matrix, adding a few

unsymmetric contributions to the problem can affect the solution time and memory consumption significantly.

If, however, the displacement field is considered as known, then a small, symmetric system of equations for $\mathbf{u}_p$ and $\mathbf{q}_p$ can be solved separately. From an implementation point of view, the `nojac()` operator is then applied to $\mathbf{u}$ in the weak formulation. This approach is a much more efficient, but requires a special solver strategy where the displacement field is solved first.

REFERENCE PLANE

When the warping displacement is evaluated with respect to a reference plane, the quadratic error R^2 in Equation 3-248 is evaluated only at the selected point:

$$R^2 = \sum_i (\mathbf{u}_i - \mathbf{u}_{a,i}) \cdot (\mathbf{u}_i - \mathbf{u}_{a,i}) \tag{3-250}$$

SYMMETRY TYPE

When a symmetry type is set as either symmetry or antisymmetry, the reference point 'p' is projected onto the symmetry plane. Then, the translation $\mathbf{u}_p$ and rotation θ_p have the following constraints for symmetry:

$$\mathbf{u}_p \cdot \mathbf{n} = 0$$
$$\theta_p \cdot \mathbf{t}_1 = 0$$
$$\theta_p \cdot \mathbf{t}_2 = 0$$

and the following constraints for antisymmetry:

$$\mathbf{u}_p \cdot \mathbf{t}_1 = 0$$
$$\mathbf{u}_p \cdot \mathbf{t}_2 = 0$$
$$\theta_p \cdot \mathbf{n} = 0$$

Definition of Warping

The warping displacement field is the difference between the total displacement field and the average rigid body displacement:

$$\mathbf{u}_w = \mathbf{u} - \mathbf{u}_a$$

The warping displacement in the normal direction is then defined as

$$u_{\mathrm{wn}} = \mathbf{u}_{\mathrm{w}} \cdot \mathbf{n}$$

where $\mathbf{n}$ is the normal to the boundary. For the case of geometric nonlinearity, the deformed (spatial) normal is used.

Energy Quantities

There are several energy quantities computed in the Structural Mechanics interfaces. This section is a summary of these quantities and how to compute in different situations.

In this section:

- Elastic Energy
- Kinetic Energy
- Dissipated Energy
- Mechanical Energy Flux
- Energy Variables

Elastic Energy

LINEAR ELASTIC MATERIALS

The elastic energy is defined as the recoverable energy stored in an elastic material or spring. The elastic strain energy density in an elastic material is defined as

$$W_s = \int_0^{\varepsilon_{el}} \sigma : d\varepsilon_{el} \tag{3-251}$$

If the linear elasticity is assumed, then

$$\sigma = \sigma_i + C : \varepsilon_{el}$$

where σ_i is the initial stress. The integration can then be carried out analytically and the result is

$$W_s = \sigma_i : \varepsilon_{el} + \frac{1}{2}(C : \varepsilon_{el}) : \varepsilon_{el} = \frac{1}{2}(\sigma_i + \sigma) : \varepsilon_{el}$$

This expression is used for the Linear Elastic Material model in a Stationary or Time Dependent analysis. An implication is that if you modify the linear elastic model in a way that violates the assumption about stress–strain linearity above, then the computed strain energy density may be wrong, for example, using a strain dependent Young's modulus or a nonconstant initial stress.

In the case of Frequency Domain analysis, only the harmonic part is considered. That is, a constant prestress does not contribute to the strain energy density. To emphasize this, the concept of *stored energy* is used. The stored energy is the cycle average of the elastic energy; that is,

$$W_h = \frac{1}{T}\int_0^T \frac{1}{2}\sigma(t):\varepsilon(t)\ dt$$

The harmonic stress and strain components are generally not in phase with each other, so the cycle average of the stored energy is computed as

$$W_h = \frac{1}{4}\mathrm{real}(\sigma:\bar{\varepsilon})$$

where the stress and strain are considered as complex quantities, and the overline denotes a complex conjugate.

In frequency domain, the strain energy density, W_s, is a complex valued quantity having the property that it acts as a potential for the instantaneous stress.

HYPERELASTIC MATERIALS

For a Hyperelastic material, the strain energy density function is the fundamental quantity from which stresses are derived. The form of the strain energy density function is determined by the hyperelastic model used.

Nonlinear Elastic Materials

For a Nonlinear Elastic Material, the strain energy density is computed in different ways depending on the material model selected. If the integration in Equation 3-251 can be performed analytically, then a closed form expression is used, similar to what is done in the linear elastic material. If not, then the integral is actually computed using the `integrate()` operator.

Structural Elements

For structural elements, the strain energy density is split into membrane, bending, and shear parts, which are then summed into a total strain energy density.

The strain energy density for all elastic domains are integrated to give a total elastic strain energy, which contains all elastic energy stored in a certain physics interface.

Elastic boundary conditions, such as Spring Foundation, Thin Elastic Layer, and Springs in joints in the Multibody Dynamics interface, also contribute to the total

elastic strain energy variable. In these cases, linearity is assumed, so if you enter nonlinear data, you will probably need to adjust the strain energy expressions.

The Multibody Dynamics interface is available with the Multibody Dynamics Module.

Kinetic Energy

For all materials, a kinetic energy density is computed. For a time-dependent study, it is defined as

$$W_k = \frac{1}{2}\rho v^2$$

where ρ is the mass density, and v is the velocity.

In Frequency Domain, the velocity is related to the displacement as $v = i\omega u$, and the kinetic energy is defined as

$$W_\mathrm{k} = \frac{1}{4}\rho\omega^2 u^2$$

which represents the cycle average.

In addition, kinetic energy variables include contributions from added mass, point masses, and moments of inertia. For structural elements like beams and shells, also distributed moment of inertia contributions are taken into account.

Dissipated Energy

There are many possible mechanisms for energy dissipation in a structure:

- Material damping (loss factor, Rayleigh damping, or viscous damping)
- Viscoelasticity
- Dissipation by plasticity or creep
- Viscous damping in boundary conditions, springs, and joints
- Friction in mechanical contact

The general form of dissipation loss is

$$Q_h = \sigma{:}\dot{\varepsilon}$$

The treatment of dissipated energy is fundamentally different depending on whether the analysis is in frequency domain or not. For stationary or time-dependent cases, the dissipated energy must be accumulated over parameter ranges or time, which means extra degrees of freedom must be added. For this reason, you must explicitly select to compute the mechanical dissipation in these cases. This is done in the **Energy Dissipation** section of the material model.

In frequency domain, the dissipated energy per cycle is computed using a closed form expression. It is always available as postprocessing variables. The expression used is

$$Q_h = \frac{1}{2}\text{real}(\sigma:\overline{(i\omega\varepsilon)})$$

In the case of Rayleigh damping, the stresses are not directly affected by the damping, since it is not part of the stress–strain relation. In a problem which is linear elastic except for the Rayleigh damping, the stresses and strains are in phase with each other, and does thus not give any damping contribution. This is handled by explicitly adding an extra contribution to dissipation.

Mechanical Energy Flux

The mechanical energy flux is a vector formed by the multiplication of the stress tensor and the velocity vector

$$\mathbf{I} = -\sigma \cdot \mathbf{v}$$

The reason for the minus sign in the definition is that if you put a pressure on an exterior boundary, and it moves in the direction of the load, then a positive power input in the direction of the load is obtained.

In the time domain, the expression above is used. In frequency domain, two versions are supplied. The *complex mechanical energy flux* is the complex vector formed by multiplying the stress tensor by the complex conjugate of the velocity:

$$\mathbf{I} = -\sigma \cdot \tilde{\mathbf{v}}$$

The *mechanical energy flux* is in the frequency domain defined as a real quantity, the cycle average of the complex mechanical energy flux.

$$\mathbf{I} = \frac{1}{2}\text{real}(-\sigma \cdot \tilde{\mathbf{v}})$$

Energy Variables

The energy variables used in the structural mechanics interfaces are summarized in Table 3-9.

TABLE 3-9: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS.

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE[1]	COMMENT
phys.Ws	Elastic strain energy density	J/m^3	solid, mbd, shell, plate, mbrn, beam, truss, wire	
phys.WsM	Membrane strain energy	J/m^2	shell, plate	Contributes to phys.Ws
phys.WsB	Bending strain energy	J/m^2	shell, plate	Contributes to phys.Ws
phys.WsS	Shear strain energy	J/m^2	shell, plate	Contributes to phys.Ws
phys.WsM	Membrane strain energy	J/m	beam	Contributes to beam.WsL
phys.WsB	Bending strain energy	J/m	beam	Contributes to beam.WsL
phys.WsS	Shear strain energy	J/m	beam	Contributes to beam.WsL
beam.WsT	Torsional strain energy	J/m	beam	Contributes to beam.WsL
phys.WsL	Strain energy density per unit length	J/m	beam, truss	phys.Ws = phys.WsL/area
phys.Ws_tot	Total elastic strain energy	J	solid, mbd, shell, plate, mbrn, beam, truss, wire	Global variable containing integration and summation of all phys.Ws contributions in a physics interface.
phys.Wh	Stored energy density	J/m^3	solid, mbd, shell, plate, mbrn, beam, truss, wire	Cycle average in frequency domain of elastic energy density.
phys.WhM	Membrane stored energy	J/m^2	shell, plate	Contributes to phys.Wh

TABLE 3-9: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS.

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE	COMMENT
phys.WhB	Bending stored energy	J/m^2	shell, plate	Contributes to phys.Wh
phys.WhS	Shear stored energy	J/m^2	shell, plate	Contributes to phys.Wh
phys.Wh_tot	Total stored energy	J	solid, mbd, shell, plate, mbrn, beam, truss, wire	Global variable containing integration and summation of all phys.Wh contributions in a physics interface.
phys.Wk	Kinetic energy density	J/m^3	solid, mbd, shell, plate, mbrn, beam, truss, wire	
phys.Wk_tot	Total kinetic energy	J	solid, mbd, shell, plate, mbrn, beam, truss	Global variable containing integration and summation of all phys.Wk contributions in a physics interface.
phys.Wp	Plastic dissipation density	J/m^3	solid	Energy density dissipated by plasticity.
phys.Wp_tot	Total plastic dissipation	J	solid	Global variable containing integration and summation of all phys.Wp contributions in a physics interface.
phys.Wc	Creep dissipation density	J/m^3	solid	Energy density dissipated by creep.

TABLE 3-9: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS.

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE	COMMENT
phys.Wc_tot	Total creep dissipation	J	solid	Global variable containing integration and summation of all phys.Wp contributions in a physics interface.
phys.Wv	Viscoelastic dissipation density	J/m^3	solid	Energy density dissipated by viscosity.
phys.Wv_tot	Total viscoelastic dissipation	J	solid	Global variable containing integration and summation of all phys.Wv contributions in a physics interface.
phys.Wdmg	Damage dissipation density	J/m^3	solid	Energy density dissipated by damage.
phys.Wdmg_tot	Total damage dissipation	J	solid	Global variable containing integration and summation of all phys.Wdmg contributions in a physics interface.
phys.Wfrac	Fracture energy density	J/m^3	solid	Energy density dissipated by phase field damage.
phys.Wfrac_tot	Total fracture energy	J	solid	Global variable containing integration and summation of all phys.Wfrac contributions in a physics interface.
phys.Wed	Dielectric dissipation density	J/m^3	solid	Piezoelectric electrical damping.

TABLE 3-9: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS.

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE[1]	COMMENT
phys.Wed_tot	Total dielectric dissipation	J	solid	Global variable containing integration and summation of all phys.Wed contributions in a physics interface.
phys.Wmd	Damping dissipation density	J/m^3	solid	Piezoelectric mechanical damping
phys.Wmd_tot	Total damping dissipation	J	solid	Global variable containing integration and summation of all phys.Wmd contributions in a physics interface.
phys.Wstb	Dielectric dissipation density	J/m^3	solid	Hourglass stabilization energy density.
phys.Wstbavg	Total dielectric dissipation	J/m^3	solid	Hourglass stabilization energy density, element average.
phys.Wstb_tot	Total dielectric dissipation	J/m^3	solid	Global variable containing integration and summation of all phys.Wstb contributions in a physics interface.
phys.I*	Mechanical energy flux, * coordinate	W/m^2	solid, mbd	
phys.Icomplex*	Complex mechanical energy flux, *coordinate	W/m^2	solid, mbd	Frequency domain only.

[1]The availability of physics interfaces is based on the license for a module.

Theory for the Shell and Plate Interfaces

The theory for the Shell and Plate interfaces is discussed in this section:

- General Theory for the Shell and Plate Interfaces
- Theory for FSDT Laminated Shell
- Theory for Hyperelastic Shells
- Results Evaluation in a Laminate
- Theory for Section Stiffness
- References for the Plate and Shell Interfaces

General Theory for the Shell and Plate Interfaces

Several topics are discussed in this section:

- Geometry and Deformation
- Strains
- Offset
- Rotation Representation
- The MITC Shell Formulation
- Initial Values and Prescribed Values
- Symmetry and Antisymmetry Boundary Conditions
- External Loads
- Stress and Strain Calculations
- Local Coordinate Systems
- Connection Between Shells and Solids
- Connection Between Shells and Beams
- Fibers in Shells and Plates

GEOMETRY AND DEFORMATION

Let $\mathbf{r}$ be the undeformed shell midsurface position, ξ^i be element local (possibly nonorthogonal) coordinates with origin in the shell midsurface, and $\mathbf{n}$ be the normal to the undeformed midsurface. The thickness of the shell is d, which can vary over the element. The local coordinates ξ^1 and ξ^2 follow the midsurface, and ξ^3 is the coordinate in the normal direction. The normal coordinate has a value of $-d/2$ on the bottom side of the element, and $+d/2$ on the top side.

The position of the deformed midsurface is $\mathbf{r} + \mathbf{u}$, and the normal after deformation is $\mathbf{n} + \mathbf{a}$. To keep the normal a unit vector requires that

$$|\mathbf{n} + \mathbf{a}| = 1 \tag{3-252}$$

Expanding Equation 3-252 gives

$$(\mathbf{n} + \mathbf{a}) \cdot (\mathbf{n} + \mathbf{a}) = \mathbf{n} \cdot \mathbf{n} + \mathbf{a} \cdot \mathbf{a} + 2\mathbf{n} \cdot \mathbf{a} = 1$$

Since $\mathbf{n}$ is a unit vector, the constraint can be written as

$$\mathbf{a} \cdot \mathbf{a} + 2\mathbf{n} \cdot \mathbf{a} = 0 \tag{3-253}$$

In a geometrically linear analysis Equation 3-252 is replaced by the simpler linearized form

$$\mathbf{n} \cdot \mathbf{a} = 0$$

since the formulation in that case assumes that

$$|\mathbf{a}| \ll 1$$

The vectors $\mathbf{r}$, $\mathbf{u}$, $\mathbf{n}$, and $\mathbf{a}$ are interpolated by the nth-order Lagrange basis functions. The basic assumption is that the position of a point within the shell after deformation has a linear dependence of the thickness coordinate, and thus is

$$\mathbf{x}(\xi^1, \xi^2, \xi^3) = \mathbf{r}(\xi^1,\xi^2) + \mathbf{u}(\xi^1,\xi^2) + \xi^3(\mathbf{n}(\xi^1,\xi^2) + \mathbf{a}(\xi^1,\xi^2))$$

The superscripts indicate contravariant indices, while subscripts indicate covariant indices.

In axial symmetry, the local curvilinear coordinates have a special interpretation: ξ^1 is the arc-length along the line on which the shell is modeled, and ξ^2 is a circumferential coordinate given by $\xi^2 = R\Theta$.

STRAINS

The in-plane Green–Lagrange strain in the local covariant components can then be written as

$$\varepsilon_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial}{\partial\xi^\alpha}(\mathbf{r} + \mathbf{u} + \xi^3(\mathbf{n} + \mathbf{a})) \cdot \frac{\partial}{\partial\xi^\beta}(\mathbf{r} + \mathbf{u} + \xi^3(\mathbf{n} + \mathbf{a})) - \frac{\partial}{\partial\xi^\alpha}(\mathbf{r} + \xi^3\mathbf{n}) \cdot \frac{\partial}{\partial\xi^\beta}(\mathbf{r} + \xi^3\mathbf{n})\right] = \gamma_{\alpha\beta} + \xi^3\chi_{\alpha\beta} + (\xi^3)^2\kappa_{\alpha\beta}$$

The indices α and β range from 1 to 2. The transverse shear strains in local covariant components are

$$\varepsilon_{3\alpha} = \varepsilon_{\alpha 3} =$$

$$\frac{1}{2}\left[\frac{\partial}{\partial \xi^\alpha}(\mathbf{r}+\mathbf{u}+\xi^3(\mathbf{n}+\mathbf{a}))\cdot(\mathbf{n}+\mathbf{a}) - \frac{\partial}{\partial \xi^\alpha}(\mathbf{r}+\xi^3\mathbf{n})\cdot\mathbf{n}\right] = \zeta_\alpha + \xi^3\omega_\alpha$$

The constitutive relation for the shell elements is a plane stress assumption, as is customary in shell theory. The strain component in the normal direction ε_{33} is thus irrelevant. The different parts of the strain tensors above can be written out as

$$\gamma_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial \mathbf{u}}{\partial \xi^\alpha}\frac{\partial \mathbf{r}}{\partial \xi^\beta} + \frac{\partial \mathbf{r}}{\partial \xi^\alpha}\frac{\partial \mathbf{u}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha}\frac{\partial \mathbf{u}}{\partial \xi^\beta}\right]$$

$$\chi_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial \mathbf{r}}{\partial \xi^\alpha}\frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha}\frac{\partial \mathbf{r}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha}\frac{\partial \mathbf{n}}{\partial \xi^\beta} + \frac{\partial \mathbf{n}}{\partial \xi^\alpha}\frac{\partial \mathbf{u}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha}\frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha}\frac{\partial \mathbf{u}}{\partial \xi^\beta}\right]$$

$$\kappa_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial \mathbf{a}}{\partial \xi^\alpha}\frac{\partial \mathbf{n}}{\partial \xi^\beta} + \frac{\partial \mathbf{n}}{\partial \xi^\alpha}\frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha}\frac{\partial \mathbf{a}}{\partial \xi^\beta}\right]$$

$$\zeta_\alpha = \frac{1}{2}\left[\frac{\partial \mathbf{r}}{\partial \xi^\alpha}\cdot\mathbf{a} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha}\cdot\mathbf{n} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha}\cdot\mathbf{a}\right]$$

$$\omega_\alpha = \frac{1}{2}\left[\frac{\partial \mathbf{n}}{\partial \xi^\alpha}\cdot\mathbf{a} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha}\cdot\mathbf{n} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha}\cdot\mathbf{a}\right]$$

In a geometrically linear analysis, the nonlinear terms (products between $\mathbf{u}$, $\mathbf{a}$, and their derivatives) disappear. In all study types, the contributions from the parts $\kappa_{\alpha\beta}$ and ω_α are ignored. They are small unless the element has an extremely high ratio between thickness and radius of curvature, in which case the errors from using shell theory are large anyway.

Note that ζ_α here is a strain component, not to be confused with the local coordinate in the normal direction or offset.

OFFSET

It is possible to model a shell with a midsurface that is not located at the meshed surface but at a certain offset from it. The offset is assumed to occur along the normal of the shell surface. In this case,

$$\mathbf{r}(\xi^1,\xi^2) = \mathbf{r}_R(\xi^1,\xi^2) + \zeta_o \mathbf{n}(\xi^1,\xi^2)$$

where $\mathbf{r}_R$ is the position of the meshed *reference surface* and ζ_o is the offset distance.

Since all geometric derivatives are computed at the mesh on the reference surface, the following type of expressions are used when evaluating the strains:

$$\frac{\partial \mathbf{r}}{\partial \xi^\alpha} = \frac{\partial \mathbf{r}_R}{\partial \xi^\alpha} + \zeta_o \frac{\partial \mathbf{n}}{\partial \xi^\alpha}$$

The degrees of freedom are located on the reference surface.

All loads and boundary conditions are assumed to be applied at the midsurface, so a force acting in the plane of the shell does not cause any bending action if there is an offset.

The numerical integration of the element is performed over the reference surface. If the shell is curved, the area of the actual midsurface and the reference surface differ. This is compensated for by multiplying the weak expressions with an area scale factor (*ASF*), defined as

$$ASF = \frac{\left|\frac{\partial \mathbf{r}}{\partial \xi^1} \times \frac{\partial \mathbf{r}}{\partial \xi^2}\right|}{\left|\frac{\partial \mathbf{r}_R}{\partial \xi^1} \times \frac{\partial \mathbf{r}_R}{\partial \xi^2}\right|}$$

Any expressions depending on the coordinates are evaluated on the shell reference surface.

For conditions applied to edges, a similar length scale factor (*LSF*) is required. It is formally defined as

$$LSF = \frac{\left|\frac{\partial \mathbf{r}}{\partial \xi^1} \cdot \mathbf{t} + \frac{\partial \mathbf{r}}{\partial \xi^2} \cdot \mathbf{t}\right|}{\left|\frac{\partial \mathbf{r}_R}{\partial \xi^1} \cdot \mathbf{t} + \frac{\partial \mathbf{r}_R}{\partial \xi^2} \cdot \mathbf{t}\right|}$$

where $\mathbf{t}$ is the tangent to the edge. For an internal edge, it is possible that there is a discontinuity in thickness or offset. In such a case, the line scale factor will be an average. Edge conditions are not well defined in such situations because the position

of the midsurface can be discontinuous. In practice, errors caused by such effects are small.

The *LSF* variable is computed from the principal curvatures, see Curvature Variables in the *COMSOL Multiphysics Reference Manual*.

ROTATION REPRESENTATION

In a geometrically linear analysis, a rotation vector is defined as

$$\theta = \mathbf{n} \times \mathbf{a}$$

In a geometrically nonlinear analysis, the rotation vector axis is defined by

$$\mathbf{e}_\theta = \frac{\mathbf{n} \times \mathbf{a}}{|\mathbf{n} \times \mathbf{a}|}$$

while the amplitude of the rotation vector is computed as

$$|\theta| = \operatorname{acos}(1 + \mathbf{n} \cdot \mathbf{a}) = \operatorname{acos}\left(1 - \frac{1}{2}\mathbf{a} \cdot \mathbf{a}\right)$$

The last equality follow from the constraint in Equation 3-253.

This representation is unique only for rotations up to 180 degrees, but since the rotation vector representation is only an output convenience, it has no impact on the analysis.

THE MITC SHELL FORMULATION

The MITC formulation (Ref. 167) does not take the strain components directly from the shape functions of the element. Instead, meticulously selected interpolation functions are selected for the individual strain components. The values of the interpolated strains are then at selected points in the element tied to the value that would be computed from the shape functions. The interpolation functions and tying points are specific to each element shape and order.

Each contribution to the virtual work of the element is numerically integrated over the reference surface while the integration in the thickness direction is performed analytically. The computation of the strain energy from transverse shear deformations

uses a correction factor of 5/6 to compensate for the difference between the assumed constant average shear strain and the true parabolic distribution.

In axisymmetry, the MITC formulation is not used. Instead, a reduced integration scheme is employed in order to avoid locking for thin shells.

FOLD LINES

In regions where the discretized surface is smooth (which is always the case for plates), the normal of the shell surface is uniquely defined. When two or more shell elements meet at an angle, each element must however keep its own normal direction. It is thus not possible to have the same set of degrees of freedom for the displacement of the normal in such a point, and this is automatically handled. The automatic search for these *fold lines* compares the normals of all boundaries sharing an edge. If the angle between the normals is larger than a certain angle (with a default of 0.001 `rad`, which is approximately 0.06 degrees) it is considered as a fold line. For a fold line, the assumption is that the angle between the shell normals remains constant. This gives

$$\mathbf{n}_j \cdot \mathbf{n}_k = (\mathbf{n}_j + \mathbf{a}_j) \cdot (\mathbf{n}_k + \mathbf{a}_k)$$

or

$$\mathbf{n}_j \cdot \mathbf{a}_k + \mathbf{a}_j \cdot \mathbf{n}_k + \mathbf{a}_j \cdot \mathbf{a}_k = 0 \tag{3-254}$$

where the values or j and k range over the number of shells elements with different normals. The third term in Equation 3-254 is relevant only in a large deformation analysis because it is nonlinear. A special case occurs when two adjacent boundaries are parallel but their normal vectors have opposite directions. In this case the special constraint

$$\mathbf{a}_k = -\mathbf{a}_j$$

is applied along their common edge.

INTEGRATION

All volume integrals over a shell element are split into a surface integral, which is performed numerically, and a thickness direction integral which is performed analytically. It is thus not possible to enter data which depend on the thickness direction. All material properties are evaluated at the reference surface. Formally this can be written as

$$\int_V f(\xi^1, \xi^2, \xi^3) dV = \int_A \int_{-d/2}^{d/2} f_1(\xi^1, \xi^2) f_2(\xi^3) d\zeta dA = \int_A f_1(\xi^1, \xi^2) F_2(d) dA$$

All functions of ζ are assumed to be of the form ζ^n. Odd powers will integrate to zero, so typically the through-thickness integration will give factors d (for the case $n = 0$) and $d^3/12$ (for the case $n = 2$). The thickness d can be a function of the position.

INITIAL VALUES AND PRESCRIBED VALUES

Because the normal vector displacements are quantities that are less intuitive than the more customary nodal rotations, it is possible to specify the prescribed values in terms of nodal rotations as well as in terms of the normal vector displacement. The representation by normal vector direction is insensitive to whether the analysis is geometrically nonlinear or not. The direction of the shell normal is prescribed in the deformed state, $\mathbf{n}_0$. The prescribed values for the actual degrees of freedom, $\mathbf{a}_0$, are internally computed as

$$\mathbf{a}_0 = \frac{\mathbf{n}_0}{|\mathbf{n}_0|} - \mathbf{n}$$

If the rotation vector input is used, and the analysis is geometrically linear, then

$$\mathbf{a}_0 = \Omega_0 \times \mathbf{n}$$

where Ω_0 is the vector of prescribed nodal rotations. This relation is fully defined only when all three components of Ω_0 are given. It is also possible to prescribe only one or two of the components of Ω_0, while leaving the remaining components free. Because it has no relevance to prescribe the rotation about the normal direction of the shell, it is only possible prescribe individual rotations in a shell local system. In this case, the result becomes one or two constraint relations between the components of $\mathbf{a}_0$. The directions are the edge local coordinate system where $\mathbf{t}_1$ is the tangent to the edge and $\mathbf{t}_2$ is perpendicular and inward from the edge, in the plane of the shell. These constraints are formulated as

$$\mathbf{t}_2 \cdot \mathbf{a}_0 = -\Omega_{01}$$

$$\mathbf{t}_1 \cdot \mathbf{a}_0 = \Omega_{02}$$

Here Ω_{0i} is the prescribed rotation around the axis $\mathbf{t}_i$.

In a geometrically nonlinear analysis, it is not possible to prescribe individual elements of the rotation vector. If only one or two components have been specified, the remaining components are set to zero. The actual degrees of freedom are then computed as

$$\mathbf{a}_0 = (\mathbf{R}(\Omega_0) - \mathbf{I})\mathbf{n}$$

where $\mathbf{R}$ is a standard rotation matrix, representing the finite rotation about the given rotation vector.

Initial velocities are always given using an angular velocity vector ω as

$$\dot{\mathbf{a}} = \omega \times (\mathbf{n} + \mathbf{a}_0)$$

SYMMETRY AND ANTISYMMETRY BOUNDARY CONDITIONS

It is possible to prescribe symmetry and antisymmetry boundary conditions. As a default, they are expressed in a shell local coordinate system. If applied to a boundary, the normal to the shell is assumed to be the normal to the symmetry or antisymmetry plane. The conditions are

$$\mathbf{u} \cdot \mathbf{n} = 0$$

$$\mathbf{a} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

for the antisymmetry case. Here $\mathbf{t}_1$ and $\mathbf{t}_2$ are two perpendicular directions in the plane of the shell.

When applied to an edge, there is a local coordinate system where $\mathbf{t}_1$ is the tangent to the edge, and $\mathbf{t}_2$ is perpendicular and in the plane of the shell. The assumption is then that $\mathbf{t}_2$ is the normal to the symmetry or antisymmetry plane. The constraints are

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the antisymmetry case.

When symmetry or antisymmetry conditions are specified in a general coordinate system with axis directions $\mathbf{e}_i$, $i = 1, 2, 3$ with $\mathbf{e}_1$ as the normal to the symmetry/antisymmetry plane the constraints are

$$\mathbf{u} \cdot \mathbf{e}_1 = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_2 \times \mathbf{n}) = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_3 \times \mathbf{n}) = 0$$

or the symmetry case and

$$\mathbf{u} \cdot \mathbf{e}_2 = 0$$

$$\mathbf{u} \cdot \mathbf{e}_3 = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_1 \times \mathbf{n}) = 0$$

for the antisymmetry case. Using a general coordinate system sometimes leads to higher accuracy, since there is no element interpolation of the constraint directions involved.

Axisymmetric Models

In axisymmetric models, the only possible symmetry plane is the one having the Z-axis as normal. In this case, you can use the Symmetry Plane boundary condition. The imposed constraint is

$$w = 0$$

EXTERNAL LOADS

Contributions to the virtual work from the external load is of the form

$$\mathbf{u}_{\text{test}} \cdot \mathbf{F} + \mathbf{a}_{\text{test}} \cdot [\mathbf{M} \times (\mathbf{n} + \mathbf{a})]$$

where the forces ($\mathbf{F}$) and moments ($\mathbf{M}$) can be distributed over a boundary or an edge or concentrated in a point. The contribution from the normal vector displacement $\mathbf{a}$ is only included in a geometrically nonlinear analysis. Loads are always referred to the midsurface of the element. In the special case of a follower load, defined by its pressure p, the force intensity is

$$\mathbf{F} = -p(\mathbf{n} + \mathbf{a})$$

For a follower load, the change in midsurface area is not taken into account, in order to be consistent with the assumption that thickness changes are ignored.

STRESS AND STRAIN CALCULATIONS

The strains calculated in the element are, as described above, the covariant tensor components. They have little significance for the user, and are internally transformed to a Cartesian coordinate system. This system can be global or element local. The stresses are computed by applying the constitutive law to the thus computed strain tensor.

Each part of the covariant strain ($\gamma_{\alpha\beta}$, $\chi_{\alpha\beta}$, ζ_α) is transformed separately. They correspond to membrane, bending, and shear action, respectively, and it is thus possible to separate the stresses from each of these actions. The membrane stress is defined as

$$\sigma_{\text{m}} = D[\gamma - \gamma_{\text{i}} - \alpha(T_{\text{m}} - T_{\text{ref}}) - \beta_h(c_{\text{m}} - c_{\text{mo,ref}})] + \frac{N_{\text{i}}}{d}$$

where D is the plane stress constitutive matrix, N_{i} are the initial membrane forces, and γ_{i} the initial membrane strains. The influence of thermal strains is included through the midsurface temperature T_{m}, and the hygroscopic swelling through the midsurface moisture concentration, c_{m}. The membrane stress can be considered as the stress at the midsurface, or as the average through the thickness.

The bending stress is defined as

$$\sigma_{\text{b}} = \frac{Dd}{2}\left[\chi - \chi_{\text{i}} - \alpha\frac{\Delta T}{d} - \beta_h\frac{\Delta c_{\text{mo}}}{d}\right] + \frac{6M_{\text{i}}}{d^2}$$

where χ_{i} is the initial value of the bending part of the strain tensor (actually: the curvature), and M_{i} are the initial bending and twisting moments. ΔT is the temperature difference between the top and bottom surface of the shell, and Δc_{mo} is

the difference in moisture concentration between the top and bottom. The bending stress is the stress at the top surface of the shell if no membrane stress is present.

The average transverse shear stress is defined as

$$\sigma_s = \frac{5}{6} \cdot 2G(\zeta - \zeta_i) + \frac{Q_i}{d}$$

where G represents the transverse shear moduli, ζ_i is the initial average shear strain, and Q_i are the initial transverse shear forces. The correction factor $5/6$ ensures that the stresses are averaged so that they correspond to the ratio between shear force and thickness. The corresponding strains ζ and ζ_i are averaged in an energy sense.

The actual in-plane stress at a certain level in the element is then

$$\sigma = \sigma_m + z\sigma_b$$

where z is a parameter ranging from -1 (bottom surface) to $+1$ (top surface). The computation of the shear stress at a certain level in the element uses the following analytical parabolic stress distribution:

$$\frac{3\sigma_s(1 - z^2)}{2}$$

The shell section forces (membrane forces, bending moments, and shear forces) are computed from the stresses as

$$N = d\sigma_m$$

$$M = \frac{d^2}{6}\sigma_b$$

$$Q = d\sigma_s$$

In certain situations, the *NMQ* formulation is used to improve performance.

LOCAL COORDINATE SYSTEMS

Boundaries

Many quantities for a shell can best be interpreted in a local coordinate system aligned to the shell surface. Material data, initial stresses and stress results are always represented in this local coordinate system. You specify the orientation of the local directions in a Shell Local System node under the **Linear Elastic Material**.

The local system for stress output coincides with the orientations defined for the material input. Stresses are also available transformed to the global coordinate system.

If a Boundary System is selected, then the orientation of the shell local system is fully defined by the boundary system. When using a boundary system, it also possible to control the orientation of the shell normal by selecting the **Reverse normal direction** checkbox.

Other types of coordinate systems are not necessarily aligned with the shell surface. For such systems, the definition of the local shell surface coordinate system is as follows:

1. The local z direction $\mathbf{e}_{z1}$ is the positive geometry normal of the shell surface.
2. The local x direction $\mathbf{e}_{x1}$ is the projection of the first direction in the material coordinate system ($\mathbf{e}_{x1}$) on the shell surface

$$\mathbf{e}_{x1} = \frac{\mathbf{e}_{x1} - (\mathbf{e}_{x1} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}}{\left|\mathbf{e}_{x1} - (\mathbf{e}_{x1} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}\right|}$$

   This projection cannot be performed if $\mathbf{e}_{x1}$ is normal to the shell. In that case, the second axis $\mathbf{e}_{x2}$ of the material system instead defines $\mathbf{e}_{x1}$ using the same procedure. Thus, if

$$\mathbf{e}_{x1} \cdot \mathbf{e}_{z1} > 0.99$$

   then

$$\mathbf{e}_{x1} = \frac{\mathbf{e}_{x2} - (\mathbf{e}_{x2} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}}{\left|\mathbf{e}_{x2} - (\mathbf{e}_{x2} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}\right|}$$

3. Finally, the second in-plane direction is generated as

$$\mathbf{e}_{y1} = \mathbf{e}_{z1} \times \mathbf{e}_{x1}$$

This procedure is followed irrespective of whether a global or a local coordinate system defines the directions.

Note the following:

- When using an isotropic material, the only effect of selecting a local coordinate system is that the definition directions of local stresses change.
- When defining orthotropic and anisotropic materials, local coordinate systems do not need to be created so that they exactly follow the shell surface. It is sufficient that the local system when projected as described above gives the intended in-plane directions.

- For shells in the *XY*-plane, and for plates, the global and local directions coincide by default.
- On curved shells, local stress components can become discontinuous if there is a location where $\mathbf{e}_{x1}$ becomes perpendicular to the shell surface.
- Avoid using the Shell Local System to enter loads or constraints at an edge or point where two or more shells which do not have common orientations meet. For prescribing edge loads, a Local Edge System is usually preferred.
- In this section, every reference to stresses is equally valid for strains.

Local Edge System

Many features, such as an edge load, allow input in an edge local coordinate system. The orthogonal local edge coordinate system directions $\mathbf{x}_l$, $\mathbf{y}_l$, and $\mathbf{z}_l$ are defined so that:

- The first direction ($\mathbf{x}_l$) is along the edge. This direction can be visualized by selecting the **Show edge directions arrows** checkbox in the **View** settings.
- The third direction ($\mathbf{z}_l$) is the same as the shell normal direction. The shell normal direction can be visualized in the result template named `Undeformed geometry`, or by adding a **Coordinate System Surface** plot and selecting the **Shell, Local System**.
- The second direction ($\mathbf{y}_l$) is in the plane of the shell and orthogonal to the edge. It is formed by the cross product of $\mathbf{z}_l$ and $\mathbf{x}_l$; $\mathbf{y}_l = \mathbf{z}_l \times \mathbf{x}_l$.

When an edge is shared between two or more boundaries, the directions may not always be unique. It is then possible to use the control **Face Defining the Local Orientations** to select from which boundary the normal direction $\mathbf{z}_l$ should be picked. The default is **Use face with lowest number**.

If the geometry selection contains several edges, the only available option is **Use face with lowest number**, since the list of adjacent boundaries would then be different for each edge. For each edge in the selection, the face with the lowest number attached to that edge is then used for the definition of the normal orientation.

RESULTS EVALUATION

For visualization and results evaluation, predefined variables include all nonzero stress and strain tensor components, *principal stresses* and *principal strains*, in-plane and out-of-plane forces, moments, and von Mises and *Tresca* equivalent stresses. It is possible to evaluate the stress and strain tensor components and equivalent stresses at an arbitrary distance from the midsurface. The parameter z_{shell} (variable name `shell.z`) is found in, for example, the **Parameters** table of the **Settings** window of a

surface plot. It can be set to a value from –1 (downside) to +1 (upside). A value of 0 means the midsurface of the shell. The default value is given in the **Default through-thickness result location** section of the Shell interface.

Stresses and strains are available both in the global coordinate system and in the shell local system as described in Local Coordinate Systems.

Using the Shell Dataset

The Shell dataset is tailored for display of results computed in the Shell interface. The purpose is to simultaneously show results on the top and bottom surfaces of the shell with a separation which matches the physics thickness. A Shell dataset with appropriate settings is generated together with the result templates, and some of the default plots make use of it.

Results such as stresses and strains which have an explicit thickness dependence will be displayed with the correct values on the respective surfaces when using a Shell dataset. Results like the degrees of freedom, which are only defined on the reference surface, will be displayed with the same value on both sides.

For thin shells, it can be difficult to see the top and bottom side. You can then manually increase the separation between the displayed top and bottom surfaces by changing the value of the **Distance** parameter in the Shell dataset.

The Shell dataset is not available in 2D, so it cannot be used with the Plate interface.

Gauss-Point Evaluation

When you evaluate stress and strain results, it is usually better to use the results at Gauss points than at the element nodes. This is particularly important when you use the MITC formulation and when there are inelastic strain contributions such as thermal expansion. You can do this by applying the gpeval() operator to the selected result quantity.

The default stress plots generated from the Shell interface show the von Mises stress at Gauss points, using an expression like `gpeval(4,shell.mises)`.

RIGID DOMAIN FOR SHELLS

The inertial properties mass (m) and moment of inertia tensor ($\boldsymbol{I}$) of a rigid shell take the finite thickness into account. They are computed as

$$m = \int_V \rho dV = \int_A \rho d \; dA$$

$$\mathbf{X}_\mathrm{M} = \frac{1}{m}\int_V \rho \mathbf{X} dV = \frac{1}{m}\int_V \rho(\mathbf{r} + \varsigma \mathbf{n}) dV = \frac{1}{m}\int_A \rho d \mathbf{r} dA$$

$$I = \int_V (((\mathbf{X} - \mathbf{X}_\mathrm{M})^T \cdot (\mathbf{X} - \mathbf{X}_\mathrm{M}))E_3 - (\mathbf{X} - \mathbf{X}_\mathrm{M}) \cdot (\mathbf{X} - \mathbf{X}_\mathrm{M})^T)\rho dV =$$

$$\int_A (((\mathbf{r} - \mathbf{X}_\mathrm{M})^T \cdot (\mathbf{r} - \mathbf{X}_\mathrm{M}))E_3 - (\mathbf{r} - \mathbf{X}_\mathrm{M}) \cdot (\mathbf{r} - \mathbf{X}_\mathrm{M})^T)\rho d \; dA +$$

$$\int_A (E_3 - \mathbf{n}\mathbf{n}^T)\rho \frac{d^3}{12} \; dA$$

where E_3 and $\mathbf{X}_\mathrm{M}$ are the identity matrix and the center of mass of the rigid domain, respectively. The last term in I is accounting for the finite thickness, and there the fact that $\mathbf{n}^T \cdot \mathbf{n} = 1$ has been used.

CONNECTION BETWEEN SHELLS AND SOLIDS

This section describes the theory and assumptions behind the multiphysics couplings Solid–Thin Structure Connection and Solid–Beam Connection. Only the shell version of the connection is described in detail, since the beam version is a direct specialization to 2D.

Solver Settings

There are three types of connections between a shell and a solid of interest:

- *Type 1 connection*: The shell connects to the solid in a thin region (having the same thickness as the shell), so that shell theory is valid on both sides. This connection is the most important from the application point of view and the most difficult to create manually.
- *Type 2 connection*: The tangent plane to the shell is perpendicular to the face of a "thick" solid, in which case the physics of the connection can, at best, be approximate.
- *Type 3 connection*: The shell acts as cladding on a solid.

The first two cases have similar physics and can be treated more or less as one case. Usually, the shell should not connect to parts on the solid boundary further away than what is represented by the shell thickness (or some other user-defined distance).

Shell Perpendicular to Solid

Figure 3-53 illustrates the first two cases. The shell edge can be part of the definition of the solid but there is no assumption about that.

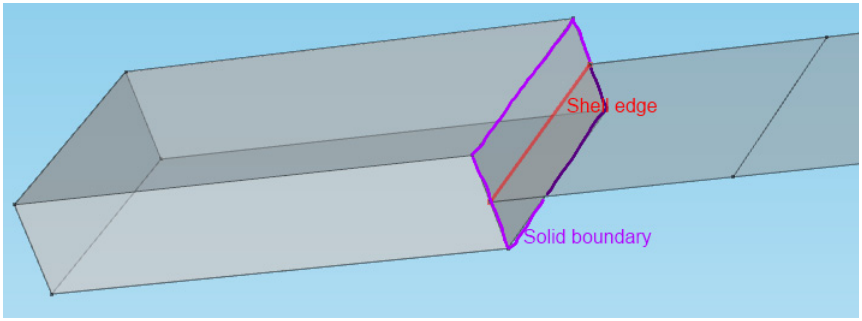


Figure 3-53: An example of a shell extending perpendicular to the solid boundary.

The connection of a solid to a shell is based on that the shell theory is valid on both sides of the connection. This can be divided into the following assumptions:

- The dimension of the solid in the shell thickness direction is the same as the thickness variable in the shell, `shell.d`.
- The midsurface of the shell to be connected to the solid is positioned at the midsurface of the solid. The reference surface of the shell can be placed anywhere.
- The cut in the solid is orthogonal to the tangent of the shell. (That is, the shell normal vector is in the plane of the cut).
- Basic shell theory assumptions are valid, for example, the direct stress in the transverse direction is negligible.

One basic shell theory assumption is actually not valid in practice: plane sections do not always remain plane under deformation. A detailed analysis shows that if there is a transverse shear force in the section, there must be a deviation from planarity to get the correct shear strain distribution. This is more important as the shell grows thicker, but without it, it is not possible to get a perfect connection. In Mindlin plate theory, shear is related to the difference between rotation and the derivative of displacement, so that plane sections remain plane, but no longer perpendicular to the midsurface. This gives

an average shear strain, while it is known from analytical solutions that the shear strain has a parabolic distribution through the thickness.

Consider a cut where the local coordinates are defined as follows:

- x' is the outward normal to the solid boundary
- z' is along the shell normal
- y' is the tangent to the shell edge, directed so that x'-y'-z' form a right-hand system

Using the plane section assumption in the shell gives the displacements in the solid as

$$u'_{\mathrm{sld}}(y',z') = u'_{\mathrm{sh}}(y',0) + z'a'_x(y',0)$$
$$v'_{\mathrm{sld}}(y',z') = v'_{\mathrm{sh}}(y',0) + z'a'_y(y',0)$$

where the subscripts **sld** and **sh** represent "solid" and "shell", respectively, and $\mathbf{a}'$ is the displacement of the shell normal, represented in the local directions.

The values on the solid boundaries should be interpreted as mapped using an extrusion operator from the shell edge.

A simple connection for the transverse direction can be generated by

$$w'_{\mathrm{sld}}(y',z') = w'_{\mathrm{sh}}(y',0)$$

This connection, however, enforces a 'plane strain condition' in the solid, which is not consistent with shell theory and which causes local unphysical stresses if Poisson's ratio is nonzero. This effect disappears within a few elements from the connection, and the approximation can, in many situations, be acceptable. This constraint is enforced if **Method** is set to **Rigid** in the multiphysics coupling.

A more accurate connection is derived in the following. The first approximation of the stress state in a moderately curved shell is

$$\begin{cases} \sigma_{x'} = A_1 + B_1 z' \\ \sigma_{y'} = A_2 + B_2 z' \\ \sigma_{z'} = 0 \\ \tau_{x'y'} = B_3 z' \\ \tau_{x'z'} = C_1(1 - z^2) \\ \tau_{y'z'} = C_2(1 - z^2) \end{cases}$$

where the relative thickness coordinate $z = \frac{2z'}{d}$ has been introduced.

Assuming that Hooke's law and plane stress conditions are valid, the transverse direct strain can be computed as

$$\varepsilon_{z'} = -\frac{\upsilon}{E}(\sigma_{x'} + \sigma_{y'}) = K_1(x',y') + 2K_2(x',y')z'$$

The transverse displacement is then, after integration with respect to z':

$$w' = K_0(x',y') + K_1(x',y')z' + K_2(x',y')z'^2$$

Thus, in order to obtain a stress-free transverse displacement, this deformation must be allowed. Note that the last term makes the solid thinner on one side of the midsurface and thicker on the other. The constant term — that is, the displacement at the midsurface of the solid — is known from the shell midsurface displacement. The other two terms depend on the stress state and thus on derivatives of the displacement, which are not readily available. The K_1 term is caused by the membrane action and the K_2 term by the bending action.

For the transverse shear stress, Hooke's law gives

$$\gamma_{x'z'} = \frac{\partial u'}{\partial z'} + \frac{\partial w'}{\partial x'} = \frac{\tau_{x'z'}}{G} = \frac{C_1}{G}(1 - z^2)$$

which can be reformulated as

$$\frac{\partial u'}{\partial z'} = \frac{C_1}{G}(1 - z^2) - \frac{\partial w'}{\partial x'} = \frac{C_1}{G}(1 - z^2) - \frac{\partial K_0}{\partial x'} - \frac{\partial K_1}{\partial x'}z' - \frac{\partial K_2}{\partial x'}z'^2$$

Note that as the K_1 term is related to membrane action, it cannot contribute to the transverse shear stress. Since the derivative in the x' direction cannot be controlled by a condition on the boundary, it is necessary to make an assumption about $u'(z')$. An integration with respect to z' gives

$$u' = M_0(x',y') + M_1(x',y')z' + M_3(x',y')z'^3$$

This shows that a third power of z' is required in order to be able to represent the correct shear strain contribution.

It is, however, not possible to directly determine the coefficients in front of the additional terms, since they depend on the actual stress state. The idea is here to

introduce them as dependent variables in the problem. These variables are defined by extra shape functions on the shell edge.

The constraints applied on the solid can then be written as

$$u'_{\mathrm{sld}}(y',z') = u'_{\mathrm{sh}}(y',0) + z'a'_x(y',0) + q_{\mathrm{u}}(y',0)(z^3 - k_{\mathrm{u}}z)d$$
$$v'_{\mathrm{sld}}(y',z') = v'_{\mathrm{sh}}(y',0) + z'a'_y(y',0)$$
$$w'_{\mathrm{sld}}(y',z') = w'_{\mathrm{sh}}(y',0) + q_{\mathrm{w1}}(y',0)zd + q_{\mathrm{w2}}(y',0)(z^2 - k_{\mathrm{w2}})d$$

Here q_{u}, q_{w1}, and q_{w2} are the new dependent variables defined on the shell edge. They are dimensionless, due to the multiplication with the shell thickness, d. The constants $k_{\mathrm{u}} = 3/5$ and $k_{\mathrm{w2}} = 1/5$ are explained below. The variable q_{w1} is proportional to the membrane axial strain, the variable q_{w2} is proportional to the bending strain, and the variable q_{u} is proportional to the transverse shear strain. Since these variables are directly related to strains, the shape function order used is one order lower than for the displacements.

If no extra equations defining q_{u}, q_{w1}, and q_{w2} are introduced, these variables try to adapt to proper values through the reaction forces on the solid. The reaction force for u' is the traction $\sigma_{x'}$ and the reaction force for w' is the traction $\tau_{x'z'}$. When taking the variation of the new dependent variables, these enforce the following constraints:

$$\mathrm{test}(q_{\mathrm{u}}) \int_{-\frac{d}{2}}^{\frac{d}{2}} (z^3 - k_{\mathrm{u}}z)\sigma_{x'}dz' = 0$$
$$\mathrm{test}(q_{\mathrm{w1}}) \int_{-\frac{d}{2}}^{\frac{d}{2}} z\tau_{x'z'}dz' = 0$$
$$\mathrm{test}(q_{\mathrm{w2}}) \int_{-\frac{d}{2}}^{\frac{d}{2}} (z^2 - k_{\mathrm{w2}})\tau_{x'z'}dz' = 0$$

The equation with q_{w1} is trivially fulfilled because the shear stress is an even function of z'. Inserting the known stress distributions gives equations that can be solved for $\mathbf{k}_u$ and $\mathbf{k}_{w2}$.

The constraint expressions must now be formulated in global directions. As a start, the constraints are written on vector form in local directions as

$$\mathbf{u}'_{sld} = \mathbf{u}'_{sh} + z'\mathbf{a}' + \mathbf{q} \tag{3-255}$$

where

$$\mathbf{q} = \begin{bmatrix} q_u(y',0)(z^3 - k_u z) \\ 0 \\ q_{w1}(y',0)z + q_{w2}(y',0)(z^2 - k_{w2}) \end{bmatrix}$$

The fact that $a_{z'} = 0$ has been used when formulating Equation 3-255.

All coordinate directions are retrieved from the shell, because the normal to the solid boundary is not necessarily constant.

The only coordinate value needed is actually z'. For the other two coordinates, only the direction is important. The coordinate in the normal direction can be computed as

$$z' = (\mathbf{X}_{sld} - \mathbf{X}_{sh}) \cdot \mathbf{N}_{sh}$$

This definition of z' assumes that the thickness of the solid does not change significantly.

Let Φ be the matrix that transforms displacements from the global system to the local system:

$$\Phi = \begin{bmatrix} \mathbf{e}_{x'} & \mathbf{e}_{y'} & \mathbf{e}_{z'} \end{bmatrix}$$

The expression for the constraints in global directions then becomes

$$\mathbf{u}_{sld} = \Phi^T \mathbf{u}'_{sld} = \Phi^T(\mathbf{u}'_{sh} + z'\mathbf{a}' + \mathbf{q}) = \mathbf{u}_{sh} + z'\mathbf{a} + \Phi^T\mathbf{q}$$

The only transformation actually needed is thus the projection of the $\mathbf{q}$ vector. For a linear case, the transformation can be written as

$$\Phi = \begin{bmatrix} s\mathbf{t}_1 \times \mathbf{N}_{sh} & s\mathbf{t}_1 & \mathbf{N}_{sh} \end{bmatrix}$$

where $\mathbf{N}$ is the undeformed shell normal (`shell.an`) and $\mathbf{t}_1$ is the shell edge tangent (`shell.tle`). The coefficient s is either 1 or −1, and it is selected so that the x' direction coincides with the outward normal of the solid.

For a geometrically nonlinear case, the corresponding deformed directions are used.

When an offset is used for the shell, it is assumed that the center of the connection is at the actual shell midsurface.

Beam Perpendicular to Solid

This is the analogous case in 2D. Beam theory assumes that the stress in the out-of-plane direction is zero. It is thus only physically sound to connect to a model where the plane stress assumption is used. The derivation above still remains valid with the following exceptions: the displacements in the local y' direction are zero and the tangent direction $\mathbf{t}_1$ is replaced by the out-of-plane direction.

The shell thickness is replaced by twice the effective radius of the beam in the equations defining the displacements.

Shell Parallel to Solid

The case where the shell is parallel to the boundary of the solid exists in two versions — *Shared* and *Parallel*.

In the shared case, the shell is modeled on a boundary which is a face of the solid. In this case, it assumed that the names of the displacement degrees of freedom in the solid and shell interfaces are not the same. If the same names are used, there is no need to use a connection feature, since the coupling is then automatic. A shell offset can be used to model an actual distance between the boundaries. For a layer "glued" on the solid, the offset would equal half the shell thickness.

In the parallel case, a separate boundary is used for modeling the shell. The distance between the shell and the face of the solid is taken into account when setting up the constraints, so that

$$\mathbf{u}_{\mathrm{sld}} = \mathbf{u}_{\mathrm{shl}} - \zeta \mathbf{a}$$

where ζ is a distance from the solid to the shell. The right-hand side is mapped from the shell to the solid using an extrusion operator. The default is that ζ is half the shell thickness, but you can also use the geometrical distance between the boundaries, or a user-defined distance.

CONNECTION BETWEEN SHELLS AND BEAMS

When connecting elements from the Shell interface with elements from the Beam interface, the following must be noted:

- The rotational degrees of freedom have different definitions in the two interfaces. The Beam interface uses a rotation vector representation, whereas the Shell interface uses the normal vector displacements.
- In general, a shell element locally has three translational, but only two rotational, degrees of freedom. The rotation around the normal is not part of the shell theory and is constrained to zero. The corresponding rotational degree of freedom in a beam should thus not be connected to the shell.

You can create the appropriate couplings by adding a Solid–Thin Structure Connection multiphysics coupling. The theory of this connection is outlined below.

Beam Edge to Shell Edge

This coupling is intended for the common situation where beams are attached along a plate to act as stiffeners. There are two variants of the coupling:

- The beam is modeled at an edge which is also an edge in the beam interface. This case is called **Shell and beam shared edges** in the **Shell–Beam Connection** node. In practice, the beam is usually placed on one side of the shell, and this offset plays an important role in the stiffness of the combined section. The offset, $\mathbf{d}_0$, can be given as a user input.
- The beam is modeled at a separate edge, representing the actual centerline. This case is called **Shell and beam parallel edges** in the **Shell–Beam Connection** node, and the closest geometrical distance between the edges directly gives $\mathbf{d}_0$. You do not need to use the same mesh on both lines. Since the constraints are formed for the shell edge, some parts of the beams could however become unconnected if the beam elements are very short when compared to the shell element size.

The displacement at the centerline of the beam can then be written in terms of the degrees of freedom in the shell as

$$\mathbf{u}_b = \mathbf{u}_s + (\mathbf{d}_0 \cdot \mathbf{n})\mathbf{a}$$

where $\mathbf{n}$ is the normal to the shell. The rotation vector in the beam can be expressed in the shell degrees of freedom as

$$\theta_b = \mathbf{n} \times \mathbf{a}$$

For a geometrically linear case, the constraint

$$\mathbf{n} \cdot \mathbf{a} = 0$$

is enforced by the shell interface for each shell boundary. This is why there are only two active rotational degrees of freedom. To avoid propagating this constraint to the beam, only those components of the beam rotation that act in the plane of the shell should be constrained. This can be expressed as

$$(\theta_b - \mathbf{n} \times \mathbf{a}) \cdot \mathbf{t}_i \qquad i = 1, 2$$

which can be simplified to

$$\theta_b \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 = 0$$
$$\theta_b \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 = 0$$

The constraints are actually formed on the shell edge, and the degrees of freedom are taken from the beam using a General Extrusion operator which maps values from the closest point on the beam to the shell.

The definitions of $\mathbf{n}$, $\mathbf{t}_1$, and $\mathbf{t}_2$ may, however, be discontinuous over a shell edge. For this reason, the constraint is formed using values from one boundary only if several boundaries share the edge. Another complication arises when the edge is a *fold line*, that is when the boundaries that meet do not have a common normal direction. On a fold line all three rotational degrees of freedom do exist in the shell and should then be connected to the corresponding degrees of freedom in the shell. In this case, also a third rotational constraint is formed.

Beam Point to Shell Boundary

This coupling is intended for the case when the beam is not in the same plane as the boundary modeled by shell theory. This case is called **Shell boundaries to beam points** in the **Shell–Beam Connection** node. Physically, this can be seen as a beam with one end welded to a plate. In order to get a correct stiffness representation of such a connection, it is necessary that the beam is connected to an area of the shell which is similar to the actual physical width of the beam. The connected area on the shell does not have to fit a boundary in the geometry. It is however necessary that the mesh size is such that there are at least three nodes within the connected area.

You can then select the region to connect using three different criteria.

The connected region is treated as rigid. The displacement of the shell is controlled by the displacement and rotation of the beam endpoint through

$$\mathbf{u}_s = \mathbf{u}_b + \theta_b \times (\mathbf{X}_s - \mathbf{X}_b)$$

The coordinates of the shell are evaluated at the reference surface.

As it is only possible to constrain the in-plane rotations of the shell, the continuity in rotation is projected onto the shell, giving

$$\theta_b \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 = 0$$
$$\theta_b \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 = 0$$

The rotation of the beam around the normal of the shell, which does not participate in the rotation constraints, is indirectly connected through the displacement equation, so it implicitly receives an appropriate stiffness.

Beam Point to Shell Edge

This case is identical to the previous case, with the only exception that the selection in the Shell interface is an edge. The beam can have any orientation relative to the shell edge. This case is called **Shell edges to beam points** in the **Shell–Beam Connection** node.

FIBERS IN SHELLS AND PLATES

Fiber reinforcement can be included in shells and plates by adding **Fiber** nodes under a Linear Elastic material. Each added **Fiber** node contributes to the stress tensor according to the chosen **Fiber placement**:

- When **Volume distribution** is chosen, the fibers are evenly distributed across the thickness of the shell with a volume fraction v_{fib} and a direction **a** (see Figure 3-54).
- When **Top surface**, **Midsurface**, **Bottom surface**, or **User defined** is chosen, the fibers lie on a surface placed at a relative offset z_{fib} measured from the shell midsurface. In this case, the volume fraction of fibers, v_{fib}, is computed from

$$v_{\text{fib}} = \frac{\pi d_{\text{fib}}}{4 s_{\text{fib}}} \frac{d_{\text{fib}}}{d}$$

where the fiber diameter d_{fib} and the spacing s_{fib} perpendicular to the fiber direction are input parameters.

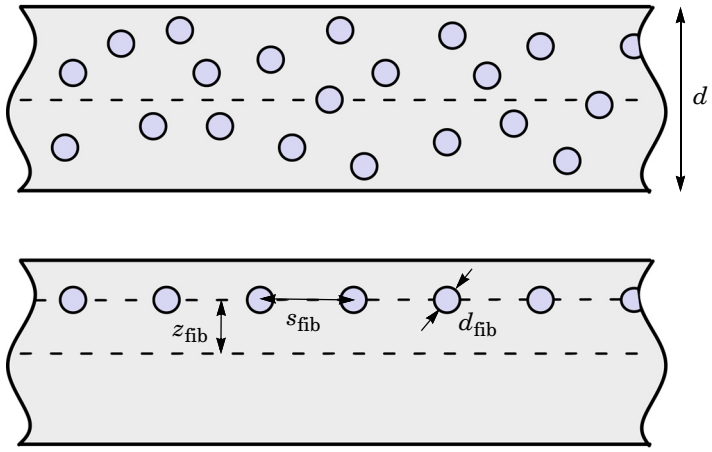


Figure 3-54: A volume distribution of fibers (top), and a fiber placement on a surface (bottom).

The elastic strain in the fiber direction

$$\varepsilon_a = \mathbf{a} \cdot \varepsilon \cdot \mathbf{a}$$

can be decomposed into a membrane part:

$$\varepsilon_{a,m} = \mathbf{a} \cdot \gamma \cdot \mathbf{a}$$

and a bending part:

$$\varepsilon_{a,b} = \mathbf{a} \cdot \chi \cdot \mathbf{a}$$

For Linear Elastic fibers placed on a surface, the definition of the membrane stress is augmented with the following contribution:

$$\sigma_{\text{fib,m}} = v_{\text{fib}}[(E_{\text{fib}} - E)\varepsilon_{a,m} + (E_{\text{fib}} - E)z_{\text{fib}}\varepsilon_{a,b}]\mathbf{a} \otimes \mathbf{a}$$

where E_{fib} is the Young's modulus of the fibers and E is the stiffness of the isotropic matrix, and $\otimes$ is the outer product of two vectors, $(\mathbf{a}\otimes\mathbf{a})_{ij} = a_i a_j$. In the same way, the definition of the bending stress is augmented with:

$$\sigma_{\text{fib,b}} = v_{\text{fib}}\left[\frac{6z_{\text{fib}}}{d}(E_{\text{fib}} - E)\varepsilon_{\text{a,m}} + (E_{\text{fib}} - E)\left(\frac{6z_{\text{fib}}^2}{d} + \frac{d_{\text{fib}}^2}{2d}\right)\varepsilon_{\text{a,b}}\right]\mathbf{a} \otimes \mathbf{a}$$

When the **Wire** approximation is selected, the term multiplying the square power of the fiber diameter is neglected. This implies neglecting the added bending stiffness due to the finite extension of the fiber in the thickness direction. The formulas for a **Volume distribution** of fibers can be obtained considering $z_{\text{fib}} = 0$ and $d_{\text{fib}} = d$.

With the **Wire** approximation, the constitutive relation of the fibers can be specified from uniaxial test data, selecting the **Uniaxial data** option for the **Material model**. In this case fibers are assumed to be placed on the shell midsurface.

See also the Distributed Fiber Models section in the *Structural Mechanics Theory* chapter.

Thermal Expansion

Thermal properties of the fiber material may differ significantly from those of the matrix material. It is therefore possible to model the thermal expansion of the fibers independently from the thermal expansion of the surrounding matrix material. Two basic assumptions are that the volume occupied by the fibers is small as compared to the base material, and that thermal expansion (or contraction) occurs in the fiber direction only (there is zero transverse thermal expansion in the fibers).

The total strain of the fibers

$$\varepsilon_{\text{a}} = \mathbf{a} \cdot \varepsilon \cdot \mathbf{a}$$

can be decomposed into a membrane part:

$$\varepsilon_{\text{a,m}} = \mathbf{a} \cdot \gamma \cdot \mathbf{a}$$

and a bending part:

$$\varepsilon_{\text{a,b}} = \mathbf{a} \cdot \chi \cdot \mathbf{a}$$

When considering thermal expansion, the stress contribution from the fibers is computed using the elastic strain:

$$\varepsilon_{\text{a,el}} = \mathbf{a} \cdot \varepsilon \cdot \mathbf{a} - \varepsilon_{\text{th,fib}}$$

The thermal strain in the fibers, $\varepsilon_{th,fib}$, can be specified by different means. When the secant coefficient of thermal expansion is used, it reads

$$\varepsilon_{th,fib} = \alpha_s(T_{fib} - T_{ref})$$

Here, the secant coefficient of thermal expansion α_s can be temperature-dependent. The reference temperature T_{ref} is the temperature at which there are no thermal strains in the fibers, and T_{fib} is the current fiber temperature.

When the tangent coefficient of thermal expansion is used, α_t, the thermal strain is given by

$$\varepsilon_{th,fib} = \exp\left(\int_{T_{ref}}^{T_{fib}} \alpha_t(\tau)d\tau\right) - 1$$

It is also possible to explicitly enter the thermal strain, dL, as a function of the fiber temperature

$$\varepsilon_{th,fib} = dL(T_{fib})$$

The coefficient of the thermal expansion or the thermal strain are considered in the fiber direction **a**, since thermal expansion is assumed to occur in the fiber direction only.

When **Thermal Bending** is active, and the **Temperature difference in thickness direction** ΔT_z is specified, the current temperature of the fibers can be obtained as

$$T_{fib} = T + \Delta T_z \frac{z_{fib}}{d}$$

If the **Temperature gradient in thickness direction** T' is specified, it is instead given by

$$T_{fib} = T + T' z_{fib}.$$

- See also Thermal Expansion in Fibers.

Theory for FSDT Laminated Shell

The theory of laminated shells is discussed in this section. The **Linear Elastic Material, Layered** node in the Shell interface allows the modeling of laminated shells, also known

as composite laminates, having different orthotropic properties per layer. The first-order shear deformation theory (FSDT) is used to find homogenized equivalent material properties of the laminate.

In this section:

- About Composite Laminates
- Equivalent Single Layer (ESL) Theory
- First-Order Shear Deformation Theory (FSDT)
- Integration in a Laminate
- Lamina Constitutive Law
- Laminate Constitutive Law
- Shear Correction Factor Computation
- Results Evaluation in a Laminate

ABOUT COMPOSITE LAMINATES

A layered shell, also known as a composite laminate, is a thin-walled structure in having many layers of different orthotropic (or optionally isotropic/anisotropic) material stacked on top of each other.

A layered shell can be active either on free surfaces embedded in 3D, or on the boundary of a solid object in 3D. In the latter case, it can be used to model a reinforcement on the surface of a solid object. Similarly, it can be active on free boundaries in 2D axisymmetry, or on the boundary of a solid object in 2D axisymmetry.

A simple form is assumed for the variation of the displacement through the thickness in order to develop a model for the deformation that is more similar to a 2D plane stress condition than to a full 3D state.

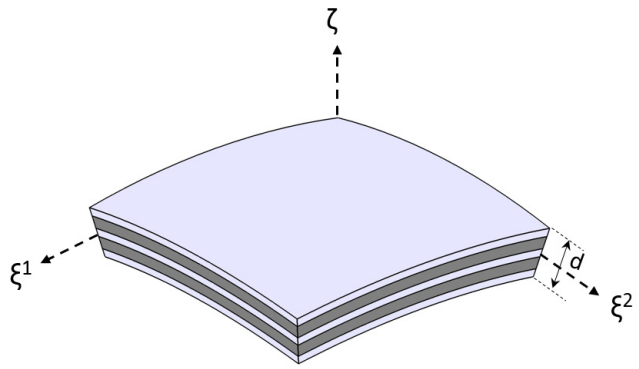


Figure 3-55: Geometry of a doubly curved composite laminate.

Figure 3-55 shows a doubly curved laminated shell with uniform total thickness, d. It is represented by orthogonal curvilinear coordinate system (ξ^1, ξ^2, ζ). The geometry representation of a layered shell is same as a single layer shell as discussed in the Geometry and Deformation section.

A typical stacking sequence of a composite laminate having n layers is shown in Figure 3-56. The thickness of each layer (d_k), as well as the fiber direction in each layer (θ_k) with respect to the first principal direction (ξ^1) of the laminate are indicated. A counterclockwise rotation of the fiber direction with respect to the ζ direction is considered as positive.

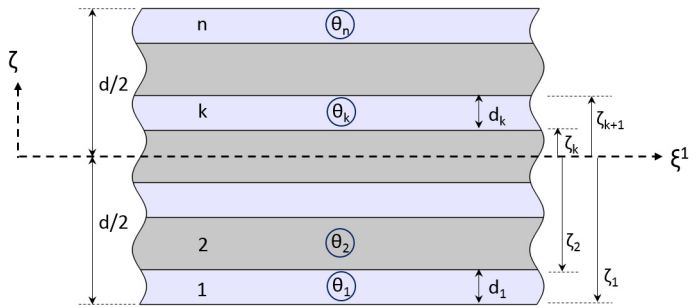


Figure 3-56: A typical stacking sequence of a composite laminate showing thickness and fiber orientation in each layer.

EQUIVALENT SINGLE LAYER (ESL) THEORY

Composite laminates are formed by stacking layers of different materials or having different fiber orientations. In general, composite laminates have a planar dimension that is orders of magnitude larger than the thickness. They are often used in the applications requiring membrane and bending strengths. Therefore, composite laminates can often be modeled using a shell element based on an equivalent single layer (ESL) theory.

In ESL theory, a heterogeneous laminated shell is converted into a statically equivalent single layer shell by reducing the 3D continuum problem to a 2D shell problem. In addition to their simplicity and low computational cost, ESL theory provides sufficiently accurate description of global response for a thin to moderately thick laminates, for example gross deflections in a laminate, critical buckling loads, and eigenfrequencies with corresponding mode shapes.

Assumptions and Restrictions

Some of the general assumptions when using shell theory to model a thin solid structure are:

- Straight lines perpendicular to the midsurface remain straight after deformation.
- Plane stress condition is assumed.
- The transverse normal strain or out-of-plane strain is nonzero, but it is not part of the formulation and is ignored while expressing the energy of the system.
- The normal rotates in such a way that it remains perpendicular to the mid surface after deformation.

The last assumption is present only in the classical laminate plate theory.

Some of the assumptions specific to ESL theory when used to model a composite laminate are:

- The material in each layer is assumed linear elastic and orthotropic.
- Each layer has a uniform thickness.
- All layers are perfectly bonded together and thus interface or delamination modeling is not possible.
- Transverse shear stresses at the top and bottom surfaces of a laminate are zero.

Classification

The ESL theories can be classified into various groups based on the description of the transverse shear stresses.

Classical Laminate Plate Theory (CLPT)

The classical laminate plate theory is an extension of Kirchhoff or classical plate theory used for single layer thin shells. In this theory, transverse shear stresses are neglected and the deformation is entirely due to the bending and in-plane stretching.

First-Order Shear Deformation Theory (FSDT)

The first-order shear deformation theory is similar to the Mindlin–Reissner shell theory used for single layer thick shells. This theory extends the kinematics of CLPT by including the gross transverse shear deformation. The transverse shear strain is assumed to be constant with respect to the thickness coordinate. As the transverse shear strain has a constant value, this theory requires a shear correction factor.

Higher-Order or Third-Order Shear Deformation Theory (HSDT)

This theory is an extension of FSDT where the displacement field is approximated in such a way that the transverse shear strain varies quadratically with respect to the thickness coordinate. It makes the transverse shear stresses zero at the top and bottom surfaces of the laminate, thus eliminating the need of a shear correction factor.

FIRST-ORDER SHEAR DEFORMATION THEORY (FSDT)

In COMSOL Multiphysics, the first-order shear deformation theory (FSDT) is implemented to model composite laminates using an equivalent single layer approach. The element used for the shell interface is of Mindlin–Reissner type, which means that transverse shear deformations are accounted for. It can thus also be used for rather

thick shells. It has an MITC (mixed interpolation of tensorial components) formulation.

The FSDT theory for a layered shell extends the ordinary theory for a single layer shell. See the following topics from the single layer shell theory for more details:

- Geometry and Deformation
- Strains
- Offset
- Rotation Representation
- The MITC Shell Formulation
- Local Coordinate Systems

FSDT differs from the single layer shell theory in the way through-thickness integrations are performed, constitutive equations are formed, and results are evaluated. In the following sections, these topics are discussed in detail.

INTEGRATION IN A LAMINATE

All volume integrals over a shell element are split into a surface integration and a through-thickness integration. Both integrations are performed numerically. The surface geometry is used for surface integration, and an extra dimension geometry is used for the through-thickness integration. It is thus possible to enter data which depend on the thickness direction.

Unlike a single layer shell, where all material properties are evaluated at the reference surface, different material and fiber orientation can be specified in each layer of the composite laminate. Formally this can be written as

$$\int_V f(\xi^1, \xi^2, \zeta) dV = \int_A \int_\zeta f_1(\xi^1, \xi^2) f_2(\zeta) d\zeta dA = \int_A \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} f_1(\xi^1, \xi^2) f_2(\zeta) dA$$

For curved laminates, an area scale factor is also included since the layers have different surface area. This is independent of whether an offset is used or not, but the offset affects the scale factor. The definition of the area scale factor (ASF) for a composite laminate is similar to that of a single layer shell.

LAMINA CONSTITUTIVE LAW

In classical laminate theory, a lamina is assumed to be in a plane stress state and all three transverse stress components (σ_{xz}, σ_{yz}, σ_{zz}) are assumed to be zero. In terms of strains, the transverse shear strain components (ε_{xz}, ε_{yz}) are zero while the transverse normal strain (ε_{zz}) is nonzero but not part of the formulation.

FSDT extends the classical laminate theory and allows nonzero transverse shear strain components.

In-Plane Constitutive Equations (Membrane and Bending Equations)

The linear constitutive relation for orthotropic lamina k in a composite laminate, can be written as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}^{(k)} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix}^{(k)}$$

where

- σ_{ij} are the stress components in the principal material directions of a lamina
- ε_{ij} are the strain components in the principal material directions of a lamina
- Q_{ij} are the elasticity matrix components in the principal material directions of a lamina

For an orthotropic lamina, the elasticity matrix components (Q_{ij}) can be defined in terms of the material constants ($E_1, E_2, \nu_{12}, \nu_{21}, G_{12}$):

$$Q_{11} = \frac{E_1}{(1-\nu_{12}\nu_{21})}$$

$$Q_{22} = \frac{E_2}{(1-\nu_{12}\nu_{21})}$$

$$Q_{12} = \frac{\nu_{12}E_2}{(1-\nu_{12}\nu_{21})} = \frac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})}$$

$$Q_{66} = 2G_{12}$$

A general composite laminate is made of several orthotropic layers, having principal material directions with different orientations with respect to the laminate coordinates (ξ^1, ξ^2, ζ) as shown in Figure 3-55. The constitutive equations for a lamina can be

transformed from its material coordinate system to the laminate coordinate system, giving:

$$\begin{bmatrix} \sigma_{\xi^1\xi^1} \\ \sigma_{\xi^2\xi^2} \\ \sigma_{\xi^1\xi^2} \end{bmatrix}^{(k)} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^1\xi^1} \\ \varepsilon_{\xi^2\xi^2} \\ \varepsilon_{\xi^1\xi^2} \end{bmatrix}$$

where

- σ_{ij} are the stress components in the laminate coordinate system
- ε_{ij} are the strain components in the laminate coordinate system
- Q_{ij} are the elasticity matrix components in the laminate coordinate system

The transformed in-plane elasticity matrix is defined as

$$\overline{Q} = T^{-1} Q T$$

The principal material direction (or fiber direction) in each lamina makes an angle (θ) with the first in-plane direction (ξ^1) of the laminate coordinate system. Hence the transformation matrix can be defined as

$$T = \begin{bmatrix} c^2 & s^2 & 2cs \\ s^2 & c^2 & -2cs \\ -sc & sc & c^2 - s^2 \end{bmatrix}$$

where $c = \cos\theta$ and $s = \sin\theta$.

Out-of-Plane Constitutive Equations (Shear Equations)

The linear constitutive relation for the transverse shear in a lamina k can be written as

$$\begin{bmatrix} \sigma_4 \\ \sigma_5 \end{bmatrix}^{(k)} = \begin{bmatrix} Q_{44} & 0 \\ 0 & Q_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}^{(k)}$$

where

- σ_{ij} are the transverse shear stress components in the principal material directions of a lamina

- ε_{ij} are the transverse shear strain components in the principal material directions of a lamina
- Q_{ij} are the elasticity matrix components in the principal material directions of a lamina

For an orthotropic lamina, the elasticity matrix components (Q_{ij}) can be defined in terms of the two shear moduli (G_{13}, G_{23}):

$$Q_{44} = G_{23}$$

$$Q_{55} = G_{13}$$

The constitutive equations of a lamina can be transformed from its material coordinate system to the laminate coordinate system and can be written as

$$\begin{bmatrix} \sigma_{\xi^2\xi^3} \\ \sigma_{\xi^1\xi^3} \end{bmatrix}^{(k)} = \begin{bmatrix} \overline{Q}_{44} & \overline{Q}_{45} \\ \overline{Q}_{45} & \overline{Q}_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^2\xi^3} \\ \varepsilon_{\xi^1\xi^3} \end{bmatrix}$$

where

- σ_{ij} are the transverse shear stress components in the laminate coordinate system
- ε_{ij} are the transverse shear strain components in the laminate coordinate system
- Q_{ij} are the elasticity matrix components in the laminate coordinate system

The transformed out-of-plane elasticity matrix is defined as

$$\overline{Q} = T^{-1} Q T$$

where the transformation matrix is defined as

$$T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

LAMINATE CONSTITUTIVE LAW

The laminate constitutive law relates the resultant membrane forces, bending moments, and transverse shear forces to the membrane strains, bending strains, and transverse shear strains.

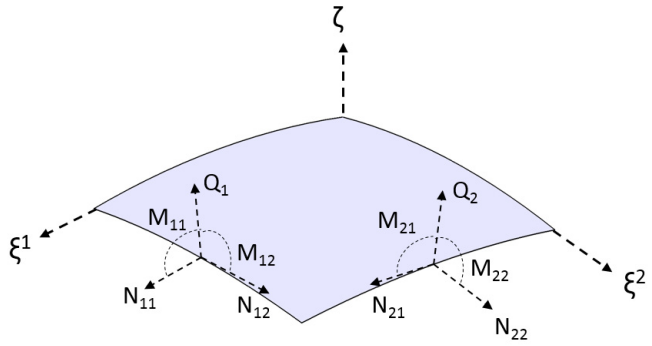


Figure 3-57: An equivalent layer of a composite laminate having n layers. The resultant membrane forces (N_{ij}), bending moments (M_{ij}), and transverse shear forces (Q_i) are shown.

The resultant membrane forces, bending moments, and transverse shear forces in a composite laminate are computed as

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \\ N_{21} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1\xi^1}\left(1+\frac{\zeta}{R_2}\right) \\ \sigma_{\xi^2\xi^2}\left(1+\frac{\zeta}{R_1}\right) \\ \sigma_{\xi^1\xi^2}\left(1+\frac{\zeta}{R_2}\right) \\ \sigma_{\xi^2\xi^1}\left(1+\frac{\zeta}{R_1}\right) \end{bmatrix}^{(k)} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \\ M_{21} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1\xi^1}\left(1+\frac{\zeta}{R_2}\right) \\ \sigma_{\xi^2\xi^2}\left(1+\frac{\zeta}{R_1}\right) \\ \sigma_{\xi^1\xi^2}\left(1+\frac{\zeta}{R_2}\right) \\ \sigma_{\xi^2\xi^1}\left(1+\frac{\zeta}{R_1}\right) \end{bmatrix}^{(k)} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^2\xi^3}\left(1+\frac{\zeta}{R_1}\right) \\ \sigma_{\xi^1\xi^3}\left(1+\frac{\zeta}{R_2}\right) \end{bmatrix}^{(k)} d\zeta$$

where

- N_{ij} are the membrane forces in the laminate coordinate system
- M_{ij} are the bending moments in the laminate coordinate system
- Q_i are the transverse shear forces in the laminate coordinate system
- σ_{ij} are the stress components in the laminate coordinate system
- K_s is the shear correction factor
- R_1 and R_2 are the principal radii of curvature of the equivalent shell (actually this implies that the laminate system is aligned with the principal curvatures, but that is of no consequence, due to the approximations made below).

For shallow shells, the following approximation can be used:

$$\left(1+\frac{\zeta}{R_1}\right) \approx 1$$
$$\left(1+\frac{\zeta}{R_2}\right) \approx 1$$

This leads to

$$N_{12} = N_{21}$$
$$M_{12} = M_{21}$$

Hence the expressions for the resultant membrane force, bending moment, and transverse shear force reduce to the following:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1\xi^1} \\ \sigma_{\xi^2\xi^2} \\ \sigma_{\xi^1\xi^2} \end{bmatrix}^{(k)} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1\xi^1} \\ \sigma_{\xi^2\xi^2} \\ \sigma_{\xi^1\xi^2} \end{bmatrix}^{(k)} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^2\xi^3} \\ \sigma_{\xi^1\xi^3} \end{bmatrix}^{(k)} d\zeta$$

This formulation, with its assumption of a moderate curvature, is the one used in the Shell interface.

The stress components can be written in terms of elasticity matrix components and strain components by using the lamina constitutive law. This establishes the relation between resultant forces and the midplane strains as given below:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon^{(0)}_{\xi^1\xi^1} + \zeta\varepsilon^{(1)}_{\xi^1\xi^1} \\ \varepsilon^{(0)}_{\xi^2\xi^2} + \zeta\varepsilon^{(1)}_{\xi^2\xi^2} \\ \varepsilon^{(0)}_{\xi^1\xi^2} + \zeta\varepsilon^{(1)}_{\xi^1\xi^2} \end{bmatrix} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon^{(0)}_{\xi^1\xi^1} + \zeta\varepsilon^{(1)}_{\xi^1\xi^1} \\ \varepsilon^{(0)}_{\xi^2\xi^2} + \zeta\varepsilon^{(1)}_{\xi^2\xi^2} \\ \varepsilon^{(0)}_{\xi^1\xi^2} + \zeta\varepsilon^{(1)}_{\xi^1\xi^2} \end{bmatrix} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \overline{Q}_{44} & \overline{Q}_{45} \\ \overline{Q}_{45} & \overline{Q}_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon^{(0)}_{\xi^2\xi^3} \\ \varepsilon^{(0)}_{\xi^1\xi^3} \end{bmatrix} d\zeta$$

The resultant force and moment expressions can be further rewritten as

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \begin{bmatrix} A_{11}\ A_{12}\ A_{16} \\ A_{12}\ A_{22}\ A_{26} \\ A_{16}\ A_{26}\ A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon^{(0)}_{\xi^1\xi^1} \\ \varepsilon^{(0)}_{\xi^2\xi^2} \\ \varepsilon^{(0)}_{\xi^1\xi^2} \end{bmatrix} + \begin{bmatrix} B_{11}\ B_{12}\ B_{16} \\ B_{12}\ B_{22}\ B_{26} \\ B_{16}\ B_{26}\ B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon^{(1)}_{\xi^1\xi^1} \\ \varepsilon^{(1)}_{\xi^2\xi^2} \\ \varepsilon^{(1)}_{\xi^1\xi^2} \end{bmatrix} \tag{3-256}$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \begin{bmatrix} B_{11}\ B_{12}\ B_{16} \\ B_{12}\ B_{22}\ B_{26} \\ B_{16}\ B_{26}\ B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon^{(0)}_{\xi^1\xi^1} \\ \varepsilon^{(0)}_{\xi^2\xi^2} \\ \varepsilon^{(0)}_{\xi^1\xi^2} \end{bmatrix} + \begin{bmatrix} D_{11}\ D_{12}\ D_{16} \\ D_{12}\ D_{22}\ D_{26} \\ D_{16}\ D_{26}\ D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon^{(1)}_{\xi^1\xi^1} \\ \varepsilon^{(1)}_{\xi^2\xi^2} \\ \varepsilon^{(1)}_{\xi^1\xi^2} \end{bmatrix} \tag{3-257}$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \begin{bmatrix} A_{44}\ A_{45} \\ A_{45}\ A_{55} \end{bmatrix} \begin{bmatrix} \varepsilon^{(0)}_{\xi^2\xi^3} \\ \varepsilon^{(0)}_{\xi^1\xi^3} \end{bmatrix} \tag{3-258}$$

where

- $\varepsilon^{(0)}_{ij}$ is the membrane part of strain
- $\varepsilon^{(1)}_{ij}$ is the bending/flexural part of strain or (curvatures)
- A_{ij} is the extensional stiffness matrix
- B_{ij} is the bending-extensional coupling stiffness matrix
- D_{ij} is the bending stiffness matrix

The extensional, bending, and coupling stiffnesses are defined as

$$A_{ij} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \overline{Q}^{(k)}_{ij} d\zeta = \sum_{k=1}^{n} \overline{Q}^{(k)}_{ij} (\zeta_{k+1} - \zeta_k)$$

$$B_{ij} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \overline{Q}^{(k)}_{ij} \zeta d\zeta = \frac{1}{2} \sum_{k=1}^{n} \overline{Q}^{(k)}_{ij} (\zeta^2_{k+1} - \zeta^2_k)$$

$$D_{ij} = \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \overline{Q}^{(k)}_{ij} \zeta^2 d\zeta = \frac{1}{3} \sum_{k=1}^{n} \overline{Q}^{(k)}_{ij} (\zeta^3_{k+1} - \zeta^3_k)$$

The in-plane laminate constitutive law, relating resultant forces and midplane strains, can also be written in matrix form as

$$\begin{bmatrix}\{N\}\\ \{M\}\end{bmatrix} = \begin{bmatrix}[A] & [B]\\ [B] & [D]\end{bmatrix}\begin{bmatrix}\{\varepsilon^{(0)}\}\\ \{\varepsilon^{(1)}\}\end{bmatrix}$$

The midplane strains can be written using resultant forces and flexibility matrices as

$$\begin{bmatrix}\{\varepsilon^{(0)}\}\\ \{\varepsilon^{(1)}\}\end{bmatrix} = \begin{bmatrix}[A] & [B]\\ [B] & [D]\end{bmatrix}^{-1}\begin{bmatrix}\{N\}\\ \{M\}\end{bmatrix}$$

$$\begin{bmatrix}\{\varepsilon^{(0)}\}\\ \{\varepsilon^{(1)}\}\end{bmatrix} = \begin{bmatrix}[a] & [b]\\ [b] & [d]\end{bmatrix}\begin{bmatrix}\{N\}\\ \{M\}\end{bmatrix}$$

where

- $[a]$ is the extensional flexibility matrix
- $[b]$ is the bending-extensional coupling flexibility matrix
- $[d]$ is the bending flexibility matrix

The stiffness and flexibility matrices are available for output, using for example **Derived Values->Point Matrix Evaluation**. The following matrix variables are defined:

- Stiffness matrix — `shell.ABD`
- Flexibility matrix — `shell.abd`
- Extensional stiffness matrix — `shell.DA`
- Extensional stiffness matrix — `shell.DA`
- Bending stiffness matrix — `shell.DD`
- Bending-extensional stiffness matrix — `shell.DB`
- Extensional flexibility matrix — `shell.Da`
- Bending flexibility matrix — `shell.Dd`
- Bending-extensional flexibility matrix — `shell.Db`

SHEAR CORRECTION FACTOR COMPUTATION

The transverse shear strains and stresses computed from the FSDT theory are averaged values. For this reason, they need shear correction factors in order to give correct strain energy contributions.

In order to compute the shear correction factor, the following assumptions are taken:

- Based on 3D elasticity theory, it is known that the first derivative of the transverse shear strain with respect to the thickness coordinate is a straight line. Thus, the transverse shear strains in each lamina are assumed to have parabolic profiles.
- In order to avoid nonlinear equations, it is assumed that there is no cross coupling between transverse shear stresses and transverse shear strains, that is, $\overline{Q}_{45} = \overline{Q}_{54} = 0$.
- It is assumed that shear modulus $G_{ij}(\zeta)$ is only a function of the thickness and does not change in the plane of laminate.

These assumptions are only satisfied exactly for a laminate with isotropic layers. For a laminate with orthotropic layers, the first assumption does not hold exactly but is approximately satisfied. The second assumption only holds for a cross-ply laminate having 0- and 90-degree layers.

For a laminate with isotropic layers, the transverse shear stresses are zero at free surfaces and they are continuous at interfaces between two layers. The through-thickness profile matches the 3D elasticity solution very closely. For a laminate with orthotropic layers, the vanishing shear stress condition is achieved at free surfaces of a laminate, while shear stress continuity at the interfaces is not guaranteed.

Based on the first assumption, the through-thickness derivative of a transverse strain component can be defined as

$$\frac{\partial \varepsilon}{\partial \zeta} = (E\zeta + F) \cdot \varepsilon_{avg}$$

Integrating the above equation in through-thickness direction, the transverse shear strain in each layer can be written as

$$\varepsilon_i = (E\zeta^2 + F\zeta + K_i) \cdot \varepsilon_{avg}$$

where

- E and F are constant for a given laminate
- K_i is the integration constant for each layer in the laminate

The transverse shear stress for each layer can be written as

$$\sigma_i = G_i \cdot (E\zeta^2 + F\zeta + K_i) \cdot \varepsilon_{\text{avg}}$$

For n number of layers in a laminate, there are $n+2$ variables (unknowns) and thus $n+2$ equations are needed to solve them. The first $n+1$ equations are the form of shear stress continuity can be written as

$$\sigma_1^{(\text{bot})} = 0$$

$$\sigma_i^{(\text{top})} = \sigma_{i+1}^{(\text{bot})}$$

$$\sigma_n^{(\text{top})} = 0$$

Here, the subscript indicates the layer index, and the superscript indicates the interface position (top or bottom) of a particular layer.

The missing equation can be expressed in the form of the average shear stress:

$$\sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} \sigma_i d\zeta = \sum_{k=1}^{n} G_i \varepsilon_{\text{avg}} \zeta_i$$

Using this set of equations, it is possible to solve for the correct distribution of shear stresses and strains. In order to compute the shear correction factor for the strain energy contribution, an additional equation is needed. It can be obtained through an energy equivalence approach between 3D elasticity and FSDT formulations.

The transverse shear strain energy based on the 3D elasticity theory is written as

$$U_s = \frac{1}{2} \int_{A} \int_{\zeta} \sigma \gamma d\zeta dA = \frac{1}{2} \int_{A} \int_{\zeta} 4 f(\zeta)^2 G(\zeta) \varepsilon_{\text{avg}}^2 d\zeta dA$$

where the shear strain distribution is defined as

$$f(\zeta) = E\zeta^2 + F\zeta + K_i$$

The transverse shear strain energy, corresponding to (ε_{13}), based on FSDT theory is written as

$$U_s = \frac{1}{2K_{13}}\int_A 2Q_1\varepsilon_{avg}dA = \frac{1}{2K_{13}}\int_A 4A_{55}\varepsilon_{avg}^2 dA$$

The shear correction factor can be obtained by equating the two energy equations:

$$K_{13} = \frac{\int_A A_{55}dA}{\int_A\int_\zeta f(\zeta)^2 G_{13}(\zeta)d\zeta dA}$$

Similarly, the shear correction factor corresponding to (ε_{23}) is

$$K_{23} = \frac{\int_A A_{44}dA}{\int_A\int_\zeta f(\zeta)^2 G_{23}(\zeta)d\zeta dA}$$

Theory for Hyperelastic Shells

Different hyperelastic material models are constructed by specifying the elastic strain energy density. Several predefined material models are included, and also the option to enter user defined expressions. See Hyperelastic Materials.

Results Evaluation in a Laminate

For visualization and results evaluation, predefined variables include all nonzero stress and strain tensor components, principal stresses and principal strains, in-plane and out-of-plane forces, moments, and von Mises and Tresca equivalent stresses.

Stresses and strains are available in the global coordinate system, laminate coordinate system, as well as in the layer local coordinate system.

LAYERED MATERIAL SLICE PLOT

It is possible to evaluate the stress and strain tensor components and equivalent stresses in each layer of a laminate using **Layered Material Slice** plot.

The through-thickness location can be set to evaluate a quantity in the middle of a layer, at an interface between two layers, top or bottom of a laminate, and so on. The top, bottom, and the middle of a laminate can be defined as

- Bottom of a laminate: -1 (relative) and 0 (physical)
- Middle of a laminate: 0 (relative) and $d/2$ (physical)
- Top of a laminate: $+1$ (relative) and d (physical)

The default value for the through-thickness location is given in the **Default through-thickness result location** section of the Layered Shell interface.

THROUGH THICKNESS PLOT

The through-thickness variation of a quantity at one or more locations on the reference surface can be plotted using a **Through Thickness** plot. In this plot, the reference surface locations can be specified through following ways:

- By selecting one or more geometric points
- By specifying the coordinate of one or more points
- By creating a cut point dataset

LAYERED MATERIAL DATASET

The **Layered Material** dataset allows the display of results in 3D solid even though the equations are solved on a 2D surface.

Using this dataset, results can either be visualized on a 3D object or on slices created in the through-thickness direction of a 3D object. The following options are available in the dataset to create slices in the through-thickness direction:

- Mesh nodes
- Interfaces
- Layer midplanes

Sometimes, when a laminate is very thin, it becomes difficult to distinguish between a surface or a solid object. In such cases it is possible to scale the through-thickness direction in the dataset for better visualization.

Theory for Section Stiffness

The Section Stiffness node in the Shell interface allows modeling of a shell by directly entering the effective stiffness or compliance matrices without specifying the

geometrical and material properties of the cross section. A typical use case is for modeling shells with homogenized properties, for example, perforated or corrugated sheets.

The background and theory of the Section Stiffness model is based on the Theory for FSDT Laminated Shell. However, instead of performing a numerical integration in the extra dimension to account for the through thickness variation of geometrical and material properties, this is assumed to have been done beforehand. Hence, you directly enter the coefficients of the **A**, **B**, and **D** matrices in Equation 3-256 to Equation 3-258 to define the effective properties of the cross section. Additionally, shear correction factors are needed so that the transverse strains and stresses give the correct contributions to the strain energy.

A consequence of this formulation is that stresses are not straightforwardly available for result evaluation, since they are not part of the formulation. Stresses can, however, be evaluated using the following equations

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \mathbf{S}_{\mathrm{N}} \begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} + \mathbf{S}_{\mathrm{M}} \begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix}$$

$$\begin{bmatrix} \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \mathbf{S}_{\mathrm{Q}} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

where matrices $\mathbf{S}_{\mathrm{N}}$, $\mathbf{S}_{\mathrm{M}}$, and $\mathbf{S}_{\mathrm{Q}}$ are user inputs. They are by default populated with coefficient that gives the stress on the top surface of the shell.

Theory for the Layered Shell Interface

The theory for the Layered Shell interface is described in this section:

- About Layered Shells
- Governing Equations

About Layered Shells

The Layered Shell interface is based on the layerwise theory of modeling a composite laminate. A layerwise (LW) theory is very similar to the traditional 3D elasticity theory where the degrees of freedom are only the displacement fields defined in the *product geometry*. The product geometry, or a domain, is defined by the reference geometric surfaces and a virtual extra dimension in the thickness direction.

The key features of the layerwise theory and a comparison with ESL theory and 3D elasticity theory are described in Laminated Shell Theories in the documentation for the Composite Materials Module.

Governing Equations

The dependent variable in the Layered Shell interface is the displacement field. The dependent variable is available in the product geometry.

The governing differential equations, constitutive relations, the definition of deformation, stresses, strains are same as in the 3D elasticity theory as described for the Solid Mechanics interface.

See Solid Mechanics Theory for more details.

GRADIENT DEFINITION

In the analysis of deformations in 3D, the deformation gradient $\boldsymbol{F}$ is defined as

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \mathbf{u} + I$$

An equivalent definition of a deformation gradient in a product geometry of a layered shell can be written as

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla_T \mathbf{u} + \frac{d\mathbf{u}}{dz}\mathbf{n}^T + I \tag{3-259}$$

where

- ∇_T is the tangential gradient operator
- z is the extra dimension thickness coordinate (varies between 0 and d)
- $\mathbf{n}$ is the positive normal direction

Equation 3-259 is exact for flat laminates. For curved laminates, the deformation gradient expression must account for the surface area of each layer. The deformation gradient in a product geometry of a curved layered shell can be written as

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \left(\nabla_T \mathbf{u} + \frac{d\mathbf{u}}{dz}\mathbf{n}^T + I\right) \cdot \left(\frac{\partial \mathbf{X}}{\partial \mathbf{X}_\mathrm{r}}\right)^{-1}$$

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}_\mathrm{r}} = I + \left(z + \frac{d}{2}(-1 + z_{\mathrm{off}})\right)\nabla_T \mathbf{n}$$

where

- $\mathbf{X}_\mathrm{r}$ are the reference surface coordinates
- z_{off} is the relative midplane offset
- d is the laminate thickness

In some applications, it is required to model variable thickness layers. This is achieved by scaling the constant thickness of the layer (d_layer) using a thickness scale factor (lsc), which could be a function of surface coordinates. The deformation gradient in a scaled product geometry of a curved layered shell can be written as

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \left(\nabla_T \mathbf{u} + \frac{1}{\mathrm{lsc}}\frac{d\mathbf{u}}{dz}\mathbf{n}^T + I\right) \cdot \left(\frac{\partial \mathbf{X}}{\partial \mathbf{X}_\mathrm{r}}\right)^{-1}$$

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}_\mathrm{r}} = I + \nabla_T\left(\left(z_s + \frac{d_s}{2}(-1 + z_{\mathrm{off}})\right)\mathbf{n}\right)$$

where

- z_s is the scaled extra dimension thickness coordinate (varies between 0 and d_s)
- d_s is the scaled laminate thickness

AREA SCALE FACTOR

For a curved laminate, the change in surface area of each layer should be accounted for while integrating the energy expressions. The area scale factor (*ASF*) for each layer of the laminate can be defined as

$$ASF = \left|\frac{\partial \mathbf{X}}{\partial \mathbf{X}_r}\right|$$

LENGTH SCALE FACTOR

For conditions applied to edges, a similar length scale factor (*LSF*) is required. It is formally defined as

$$LSF = \frac{\left|\frac{\partial \mathbf{r}}{\partial \xi^1}\cdot\mathbf{t} + \frac{\partial \mathbf{r}}{\partial \xi^2}\cdot\mathbf{t}\right|}{\left|\frac{\partial \mathbf{r}_R}{\partial \xi^1}\cdot\mathbf{t} + \frac{\partial \mathbf{r}_R}{\partial \xi^2}\cdot\mathbf{t}\right|}$$

where $\mathbf{t}$ is the tangent to the edge. For an internal edge, it is possible that there is a discontinuity in thickness or offset. In such a case, the line scale factor will be an average. Edge conditions are not well defined in such situations because the position of the midsurface can be discontinuous. In practice, errors caused by such effects are small.

The *LSF* variable is computed from the principal curvatures, see Curvature Variables in the *COMSOL Multiphysics Reference Manual*.

INTEGRATION

All volume integrals over a layered shell element are split into a surface integration and a through-thickness integration. Both integrations are performed numerically. The surface geometry is used for surface integration and the extra dimension geometry is used for the through-thickness integration. It is thus possible to enter data which explicitly depends on the thickness direction.

Unlike a single layer shell, where all material properties are evaluated at the reference surface, different material and fiber orientation can be specified in each layer of the composite laminate. Formally this can be written as

$$\int_V f(\xi^1, \xi^2, \zeta) dV = \int_A \int_\zeta f_1(\xi^1, \xi^2) f_2(\zeta) d\zeta dA = \int_A \sum_{k=1}^{n} \int_{\zeta_k}^{\zeta_{k+1}} f_1(\xi^1, \xi^2) f_2(\zeta) dA$$

As discussed in the previous section, the area scale factor (*ASF*) is included for curved laminates since the layers have different surface area. This is independent of whether an offset is used or not, but the offset affects the scale factor.

The layer thickness scale factor (lsc) is also accounted in the integrations when variable thickness layers are present in the model.

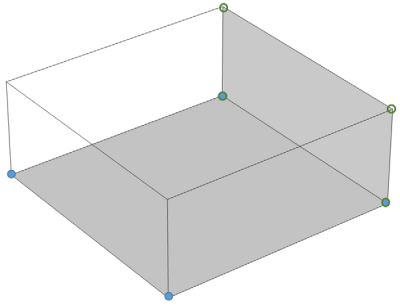
DISCRETIZATION

The Layered Shell interface can use different shape orders for the displacement field in the reference surface and in the through-thickness direction. The shape orders used can be divided into categories:

- Same shape order in the reference surface and in the extra dimension (Figure 3-58 and Figure 3-59)
- Different shape orders in the reference surface and in the extra dimension (Figure 3-60)

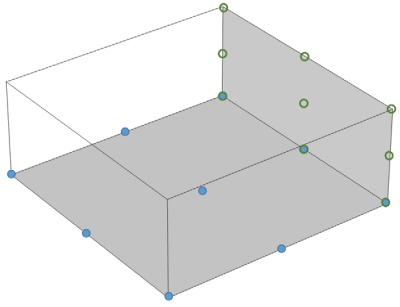
Based on the above two categories, 9 different elements are available:

- Linear (8 nodes)
- Quadratic Lagrange (27 nodes)
- Quadratic Serendipity (24 nodes)
- Cubic Lagrange (64 nodes)
- Cubic Serendipity (48 nodes)
- Quadratic-Linear Lagrange (18 nodes)
- Quadratic-Linear Serendipity (16 nodes)
- Quadratic-Cubic Lagrange (36 nodes)
- Quadratic-Cubic Serendipity (32 nodes)

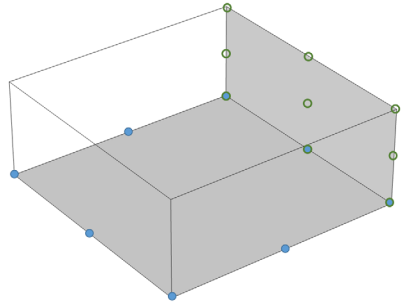


Linear (8 nodes)

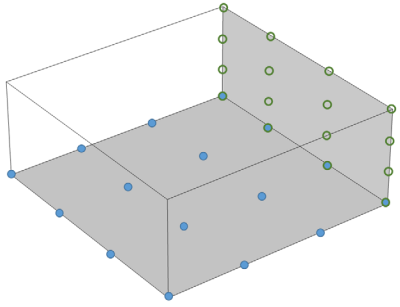
Figure 3-58: Element having linear shape order.



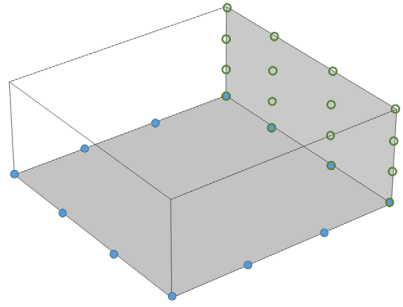
Quadratic Lagrange (27 nodes)



Quadratic Serendipity (24 nodes)



Cubic Lagrange (64 nodes)



Cubic Serendipity (48 nodes)

Figure 3-59: Elements having quadratic and cubic shape orders.

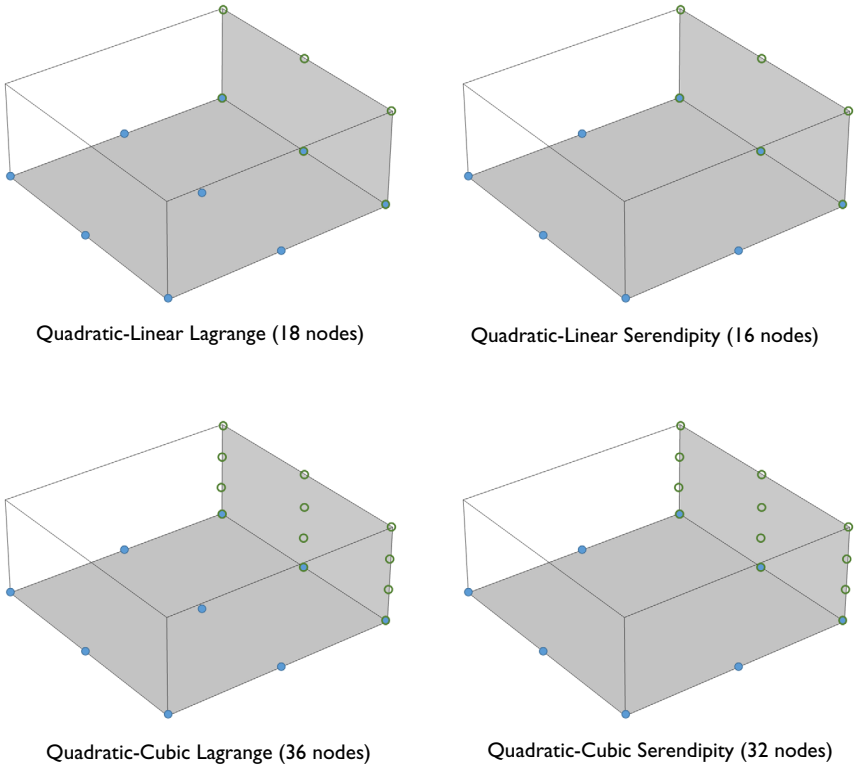


Figure 3-60: Elements having different shape orders on the reference surface and in the through-thickness direction.

FOLD-LINE CONNECTION

When two or more layered shell surfaces meet at an angle, the displacement field cannot be same on the side faces of all the layered shell surfaces. Thus, a slit condition is needed on the common edges and a connection needs to be established between the displacement field of different layered shell surfaces sharing an edge.

This is automatically handled in COMSOL Multiphysics. The automatic search for these *fold lines* compares the normals of all the layered shell surfaces sharing an edge. If the angle between the normals is larger than a certain angle (default 3°) it is considered as a fold line.

In order to connect the displacement field of different shell surfaces meeting at a fold line, the following two conditions are needed:

- The displacement at the reference surface position should be continuous. This can be written as

$$\mathbf{u}_{bj} = \mathbf{u}_{bk}$$

where $\mathbf{u}_b$ is the displacement vector at the reference surface location in the through-thickness direction.

- The angle between the shell normals remains constant. This can be written as

$$\mathbf{n}_j \cdot \mathbf{n}_k = \mathbf{u}_{rj} \cdot \mathbf{u}_{rk}$$

$$\mathbf{u}_r = \left(z + \frac{d}{2}(-1 + z_{\text{off}})\right)\mathbf{n} + \mathbf{u} - \mathbf{u}_b$$

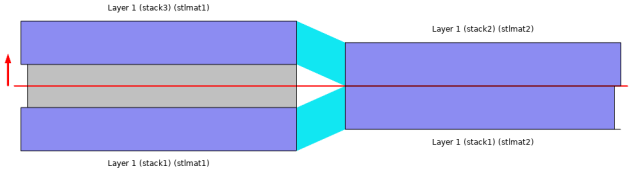
where $\mathbf{u}_r$ is the displacement vector in the through-thickness direction relative to the displacement vector at the reference surface location.

LAYERED SHELL CONTINUITY

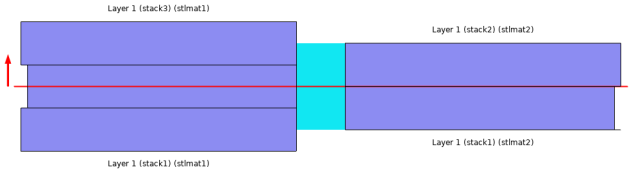
Different layered materials have their own virtual extra dimension geometry and that is why, by default, two layered materials sharing a common edge in the geometry do not have a continuous displacement field. Thus, when two layered shell surfaces having different layered materials meet side-by-side, a displacement continuity needs to be established in the through-thickness direction on the common edge.

For an example, a layered material having 3 layers shares a common edge with another layered material having 2 layers. For that case, some of the ways you can connect the two layered materials in the through-thickness direction are as follows:

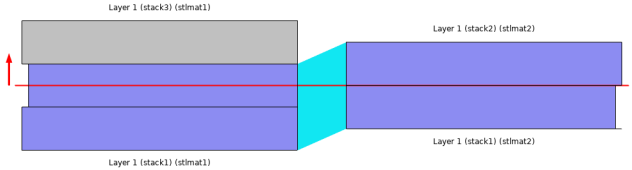
- Connecting the first and third layer of the left layered material to both the layers of the right layered material. This type of connection is typically used in a ply dropping scenario.



- Connecting the geometrically common area of both the layered materials in through-thickness direction.



- Connecting the first two layers of the left layered material to both the layers of the right layered material.



Note that continuity between different layered materials meeting side by side is correctly established for surfaces having the same normal direction at the common edge.

In case the two layered materials have normal orientation in opposite directions, you may want to switch the **Connection type** from **Straight** to **Twisted** in the **Connection Settings** section of the **Continuity** node in order to connect points in extra dimension which are geometrically close to each other.

Using the continuity condition when the two surfaces meet at an angle may give rise to the stress concentration near the common edge, as well as a locking of the global rotation of the layered shell surfaces.

Theory for the Membrane Interface

Introduction

A membrane is similar to a shell, but it has only translational degrees of freedom and the strain is constant in the thickness direction.

The thickness of the membrane is d, which can vary over the element. The displacements are interpolated by Lagrange or Serendipity shape functions.

A 2D axisymmetric membrane is similar to the 3D membrane and it has a nonzero circumferential strain in the out-of-plane direction.

Local Coordinate Systems

BOUNDARIES

Many quantities for a membrane can best be interpreted in a local coordinate system aligned to the membrane surface. Material data, initial stresses-strains, and constitutive laws are always represented in the local coordinate system.

This local membrane surface coordinate system is defined by the boundary coordinate system ($\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$).

The quantities like stresses and strains are also available as results in the global coordinate system after a transformation from a local (boundary) system.

LOCAL EDGE SYSTEM

Many features, such as an edge load, allow input in an edge local coordinate system. The orthogonal local edge coordinate system directions $\mathbf{x}_l$, $\mathbf{y}_l$, and $\mathbf{z}_l$ are defined so that:

- The first direction ($\mathbf{x}_l$) is along the edge. This direction can be visualized by selecting the **Show edge directions arrows** checkbox in the **View** settings.
- The third direction ($\mathbf{z}_l$) is the same as the membrane normal direction of the adjacent boundary.
- The second direction ($\mathbf{y}_l$) is in the plane of the shell and orthogonal to the edge. It is formed by the cross product of $\mathbf{z}_l$ and $\mathbf{x}_l$; $\mathbf{y}_l = \mathbf{z}_l \times \mathbf{x}_l$.

The local edge system can be visualized by plotting the components of the local edge transformation matrix with an **Arrow Line** plot. The matrix components are defined per

feature. For instance, the variable name for the xx-component is `<interface>.<feature_tag>.TleXX`.

When an edge is shared between two or more boundaries, the directions may not always be unique. It is then possible to use the control **Face Defining the Local Orientations** to select from which boundary the normal direction $\mathbf{z}_l$ should be picked. The default is **Use face with lowest number**.

If the geometry selection contains several edges, the only available option is **Use face with lowest number**, since the list of adjacent boundaries would then be different for each edge. For each edge in the selection, the face with the lowest number attached to that edge is then used for the definition of the normal orientation.

Strain–Displacement Relation

The kinematic relations of the membrane element are first expressed along the global coordinate axes. The strains are then transformed to the element local direction. Since the membrane is defined only on a boundary, derivatives in all spatial directions are not directly available. This makes the derivation of the strain tensor somewhat different from what is used in solid mechanics.

The deformation gradient F is in general defined as the gradient of the current coordinates with respect to the original coordinates:

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = I + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$$

In the Membrane interface, a tangential deformation gradient is computed as

$$F_t = I - \mathbf{N} \otimes \mathbf{N} + \nabla_t \mathbf{u}$$

Here $\nabla_t \mathbf{u}$ is the displacement gradient computed using the tangential gradient on the membrane surface, $\mathbf{N}$ is the normal vector to the undeformed membrane, and $\otimes$ is the outer product of two vectors, $(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j$. The tangential deformation gradient F_t then contains information about the stretching in the membrane plane.

Since the tangential deformation gradient does not contain any information about the transverse (out-of-plane) stretch λ_n, it must be augmented by the normal deformation gradient F_n to define the full deformation gradient. It is given by

$$F_n = \lambda_n \mathbf{n} \otimes \mathbf{N} = (1 + u_{nn}) \mathbf{n} \otimes \mathbf{N}$$

where $\mathbf{n}$ is the normal vector to the deformed membrane. For anisotropic materials, the shear deformation gradient $\boldsymbol{F}_\mathrm{s}$ is also needed to define the full deformation gradient. It is given by

$$\boldsymbol{F}_\mathrm{s} = (u_{1\mathrm{n}}\mathbf{t}_1 + u_{2\mathrm{n}}\mathbf{t}_2) \otimes \mathbf{N}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are the tangent vectors on the deformed membrane surface. The full deformation gradient $\boldsymbol{F}$ is then computed from the sum of tangential, shear, and normal deformation

$$\boldsymbol{F} = \boldsymbol{F}_\mathrm{t} + \boldsymbol{F}_\mathrm{s} + \boldsymbol{F}_\mathrm{n}$$

Note that $\boldsymbol{F}_\mathrm{s}$ is only nonzero for anisotropic materials, otherwise $\boldsymbol{F}_\mathrm{s} = 0$.

The right Cauchy–Green tensor $\boldsymbol{C}$ is then defined as

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$$

The Green–Lagrange strains are computed using the standard expression

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$$

The local tangential strains are calculated by transforming this tensor into the local coordinate system.

The Jacobian J is the ratio between the current volume and the initial volume. In full 3D it is defined as

$$J = \det(\boldsymbol{F}) = \sqrt{\det(\boldsymbol{C})}$$

and the area scale factor is computed from

$$J_\mathrm{A} = \frac{J}{\lambda_\mathrm{n}}$$

In the case of geometrically linear analysis, a linearized version of the strain tensor is used.

Constitutive Relation and Weak Contributions

The constitutive relations for the membrane on the reference surface are similar to those used in the Solid Mechanics interface.

The thermal strains and initial stresses-strains (only for the in-plane directions of the membrane) are added in the constitutive relation in a similar manner as it is done in Solid Mechanics.

The weak expressions in the Membrane interface are similar to that of linear elastic continuum mechanics.

See also Analysis of Deformation in the documentation of the Solid Mechanics interface.

Wrinkling

Due to the zero bending stiffness assumption, the numerical treatment of thin structures is much simpler with the membrane theory as compared to shell theory. However, this assumption is disadvantageous in some cases such as wrinkling, which happens when the membrane is subjected to negative principal stresses.

A thin shell will wrinkle when the compressive stress reaches a critical level defined by its bending stiffness, which is a local buckling phenomenon. When such thin structures are modeled within the membrane theory, wrinkles appear at the onset of the compressive stresses as the bending stiffness is assumed to be zero. Due to zero bending stiffness such states can be represented by continuously distributed infinitesimal wrinkles.

When using the traditional membrane theory, which does not incorporates a wrinkling model, negative principal stresses result in an equilibrium instability. In order to overcome this instability, the wrinkling model within the framework of the tension field theory can remove compressive stresses from wrinkled regions resulting in a correct stress distribution, Ref. 169–171.

The modified membrane theory, which incorporates a wrinkling model to the traditional membrane theory, disregards the out-of-plane deformation in wrinkling; so wrinkles including details like their amplitude and wavelength are determined on the mean surface of the membrane (Ref. 169–171).

There are two approaches to embed the tension field theory into the framework of classical membrane theory, but essentially both approaches are equivalent (Ref. 171). One approach is to replace the strain energy density with a relaxed variant of it, while the other approach is to modify the deformation gradient (Ref. 169). While the first

approach can only be used for isotropic membranes, the second approach is more general (Ref. 169), and it is the method implemented in COMSOL Multiphysics.

The undeformed configuration of the membrane is described by the material coordinates X, Y, and Z. The uniaxial stress occurs in the X direction; Y is the wrinkling direction and Z is the normal to the membrane plane (Figure 3-61). The deformed configuration is represented by the x, y, and z coordinates. After deformation, the membrane can be in one of three possible states:

- A taut state, which is achieved when both in-plane principal stresses are positive, and there is no need to modify the deformation tensor.
- A wrinkled state, which happens when one of the principal stresses is negative. A modification to the deformation gradient tensor is necessary.
- A slack state, reached when both in-plane principal stresses are negative. In this case, the stress tensor is set to be zero.

As shown in Figure 3-61 there are three different kinematic descriptions (Ref. 169):

- The tensor $\boldsymbol{F}^*$ maps the reference configuration to the true wrinkled configuration. This mapping is not suitable to describe the strain field in wrinkled membranes.
- The tensor $\boldsymbol{F}$ maps the reference configuration to the mean configuration, where the current area is smaller than the actual wrinkled area. Hence, this mapping is also not suitable to describe the strain field in wrinkled membranes.
- The tensor $\bar{\boldsymbol{F}}$ maps the reference configuration to a fictitious lengthened configuration, where the current area is equal to the actual wrinkled area. This mapping is suitable to describe the strain field in wrinkled membranes.

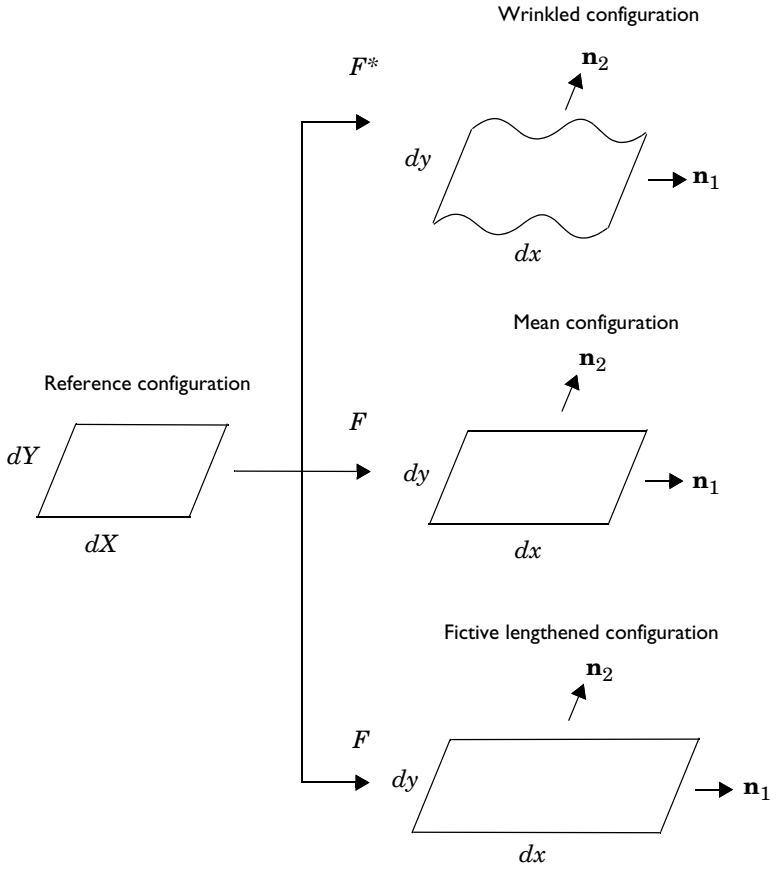


Figure 3-61: Kinematics of wrinkling.

When $\mathbf{n}_1$ is the direction of uniaxial extension, and assuming that wrinkling occurs in the $\mathbf{n}_2$ direction, the modified deformation tensor $\bar{F}$ is written as

$$\bar{F} = (I + \beta \mathbf{n}_2 \otimes \mathbf{n}_2)F \tag{3-260}$$

Here, $\otimes$ is the outer product of two vectors, $(\mathbf{a}\otimes\mathbf{b})_{ij} = a_i b_j$, and β is the wrinkling parameter, so $\beta = 0$ represents a taut condition. According to the orthogonality condition in tension field theory, these two vectors satisfy

$$\begin{aligned} \mathbf{n}_1 \cdot \sigma \cdot \mathbf{n}_2 &= 0 \\ \mathbf{n}_2 \cdot \sigma \cdot \mathbf{n}_2 &= 0 \end{aligned} \tag{3-261}$$

where σ is the Cauchy stress tensor written in terms of the second Piola–Kirchhoff stress tensor

$$\sigma = \frac{1}{J}\bar{F}S(\bar{F})\bar{F}^T$$

Assuming that the mean configuration F is known, the only unknowns in Equation 3-260 and Equation 3-261 are β and $\mathbf{n}_2$.

Since the material properties and membrane theory itself are given in the reference configuration, Equation 3-260 and Equation 3-261 are transformed to the reference configuration (Ref. 170). The fictive Green–Lagrange strain tensor $\bar{E}$ is then written with the help of a vector in the reference configuration which corresponds to the wrinkling direction $\mathbf{n}_2$, so $\mathbf{g} = \mathbf{n}_2 \cdot F$,

$$\bar{E} = \frac{1}{2}(\bar{F}^T\bar{F} - I) = E + \left(\beta + \frac{\beta^2}{2}\right)F^T\mathbf{n}_2 \otimes \mathbf{n}_2 F$$

This can be written as

$$\bar{E} = E + \beta_m \mathbf{N}_2 \otimes \mathbf{N}_2$$

where β_m and $\mathbf{N}_2$ are the wrinkling parameter and wrinkling direction in the reference configuration. The two unknowns β and $\mathbf{n}_2$ in the deformed configuration are replaced by the two unknowns β_m and $\mathbf{N}_2$ in the reference configuration.

The membrane surface is spanned by a coordinate system having two in-plane orthogonal unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$. Thus, $\mathbf{N}_1$ and $\mathbf{N}_2$ can be written with the help of the wrinkling angle α_m

$$\begin{aligned} \mathbf{N}_1 &= \cos(\alpha_m)\mathbf{e}_1 + \sin(\alpha_m)\mathbf{e}_2 \\ \mathbf{N}_2 &= -\sin(\alpha_m)\mathbf{e}_1 + \cos(\alpha_m)\mathbf{e}_2 \end{aligned}$$

So the scalar wrinkling angle α_m is sufficient to define the wrinkling vector $\mathbf{N}_2$. The two nonlinear equations in Equation 3-261 are then solved for the two unknowns α_m and β_m with the Newton–Raphson method. Once the parameters α_m and β_m are determined, the deformation gradient and the second Piola–Kirchhoff stress tensor are modified to get the correct stress distribution and to define the wrinkled regions.

External Loads

The contributions to the virtual work from an external load is given by

$$\mathbf{u}_{\text{test}} \cdot \mathbf{F}$$

where the force $\mathbf{F}$ can be distributed over a boundary, an edge, or it can be concentrated in a point. In the special case of a follower load, defined by a pressure p, the force intensity is $\mathbf{F} = -p\mathbf{n}$, where $\mathbf{n}$ is the normal vector to the membrane surface in the deformed configuration.

For a follower load, the change in midsurface area is taken into account, and integration of the load is done in the deformed configuration.

Stress Calculations

The stresses are computed by applying the constitutive law to the computed strains.

Membranes do not support transverse nor bending forces, and the only section forces are defined as

$$N = d \cdot \bar{s}$$

Here, $\bar{s}$ is the local stress tensor which contains in-plane stress components only.

Theory for the Beam Interface

The Beam Interface theory is described in this section:

- Shape Functions
- Geometric Nonlinearity
- Computation of Stresses
- Stress–Strain Relation
- Effects of Thermal Expansion
- Hygroscopic Swelling
- Initial Stress and Strain
- Deformation Measures
- Theory for Section Stiffness
- Stress Evaluation
- Common Cross Sections

Beam Cross Section

Shape Functions

The beam element has different shape functions for representing the displacements in different directions.

- The axial extension is represented by a linear shape function.
- The twist around the beam axis (3D only) is represented by a linear shape function.
- The bending displacement and corresponding rotation are represented by cubic shape functions, usually called Hermitian shape functions. These will supply exact solutions to the underlying beam equations as long as distributed loads do not vary with position.

The shape functions for bending depend on whether Timoshenko theory is employed or not.

In the beam local system, the displacements, $\mathbf{u}$, and rotations, θ, are interpolated as

$$\begin{Bmatrix} \mathbf{u} \\ \theta \end{Bmatrix} = [\mathbf{N}] \begin{Bmatrix} \mathbf{u}_1 \\ \theta_1 \\ \mathbf{u}_2 \\ \theta_2 \end{Bmatrix}$$

where the subscript refers to the two nodes of the element, and **N** is a matrix of shape functions.

$$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & N_5 & 0 & N_4 & 0 & 0 & 0 & N_6 \\ 0 & 0 & N_3 & 0 & -N_5 & 0 & 0 & 0 & N_4 & 0 & -N_6 & 0 \\ 0 & 0 & 0 & M_1 & 0 & 0 & 0 & 0 & 0 & M_2 & 0 & 0 \\ 0 & 0 & M_3 & 0 & M_5 & 0 & 0 & 0 & M_4 & 0 & M_6 & 0 \\ 0 & -M_3 & 0 & 0 & 0 & M_5 & 0 & -M_4 & 0 & 0 & 0 & M_6 \end{bmatrix}$$

The shape functions for the Euler–Bernoulli case are expressed in the local coordinate ξ, ranging from 0 to 1, as

$$N_1 = 1 - \xi \qquad N_2 = \xi$$
$$N_3 = 1 - 3\xi^2 + 2\xi^3 \quad N_4 = 3\xi^2 - 2\xi^3 \quad N_5 = L(\xi - 2\xi^2 + \xi^3) \quad N_6 = L(-\xi^2 + \xi^3)$$
$$M_1 = 1 - \xi \qquad M_2 = \xi$$
$$M_3 = -\frac{6}{L}(\xi - \xi^2) \qquad M_4 = \frac{6}{L}(\xi - \xi^2) \quad M_5 = 1 - 4\xi + 3\xi^2 \qquad M_6 = -2\xi + 3\xi^2$$

where L is the length of the beam element.

For the Timoshenko case, the shape functions are modified, so that they depend on the degree of shear flexibility. Define

$$\Phi_2 = \frac{12EI_{zz}}{G\kappa_y AL^2} = \frac{24(1+\nu)I_{zz}}{\kappa_y AL^2}$$
$$\Phi_3 = \frac{12EI_{yy}}{G\kappa_z AL^2} = \frac{24(1+\nu)I_{yy}}{\kappa_z AL^2}$$

which represent the ratios between bending and shear stiffness in the two principal directions. The shape functions are then modified so that

$$N_3^i = \frac{(N_3 + \Phi_i \hat{N}_3)}{1 + \Phi_i} \qquad N_4^i = \frac{(N_4 + \Phi_i \hat{N}_4)}{1 + \Phi_i}$$

$$N_5^i = \frac{(N_5 + \Phi_i \hat{N}_5)}{1 + \Phi_i} \qquad N_6^i = \frac{(N_6 + \Phi_i \hat{N}_6)}{1 + \Phi_i}$$

$$M_5^i = \frac{(M_5 + \Phi_i \hat{M}_5)}{1 + \Phi_i} \qquad M_6^i = \frac{(M_6 + \Phi_i \hat{M}_6)}{1 + \Phi_i}$$

where

$$\hat{N}_3 = 1 - \xi \qquad \hat{N}_4 = \xi$$

$$\hat{N}_5 = \frac{L}{2}(\xi - \xi^2) \qquad \hat{N}_6 = -\frac{L}{2}(\xi - \xi^2)$$

$$\hat{M}_5 = 1 - \xi \qquad \hat{M}_6 = \xi$$

The superscript i indicates that the shape functions for bending are no longer the same in the two principal directions. The shape functions with $i = 2$ are used for bending in the local y direction, and the shape functions with $i = 3$ are used for bending in the local z direction.

Geometric Nonlinearity

You can use the beam interface for modeling problems with large displacements and rotations, but small strains. A so-called co-rotational formulation is used. The displacement of each individual beam element is decomposed into a rigid body translation and rotation, and a local response of the rotated element which is linear.

The assumption that the individual element behaves linearly implies that you must use a fine mesh if the curvature of the deformed beam is large. The difference in rotation between the endpoints of the individual element must not be larger than it would be possible to analyze it using linear theory.

Different coordinate systems are needed for describing the beam configurations. The initial configuration of the beam can be described by a triad of orthogonal unit vectors $\mathbf{r}_i^0$. The first vector is parallel to the beam, and the second and third vectors point in the local y and z directions, respectively. The origin of the local system is taken to be the midpoint of the element. This system translates and rotates with the rigid body motion of the beam, and the new directions of the axes are called $\mathbf{r}_i$.

The rotation of the beam is represented by rotation vectors, θ. The rotation at the midpoint is approximated as the arithmetic mean of the rotations at the nodes,

$$\theta_M = \frac{\theta_1 + \theta_2}{2}$$

The rigid rotation is then represented by a rotation matrix $\mathbf{R}_r$, corresponding to this midpoint rotation. It is given by

$$\mathbf{R}_r = \mathbf{I} + \frac{\sin|\theta_M|}{|\theta_M|}\Theta_M + \frac{1 - \cos|\theta_M|}{|\theta_M|^2}\Theta_M^2$$

where Θ_M is the skew symmetric representation of the midpoint rotation vector.

$$\Theta_M = \begin{bmatrix} 0 & -\theta_{Mz} & \theta_{My} \\ \theta_{Mz} & 0 & -\theta_{Mx} \\ -\theta_{My} & \theta_{Mx} & 0 \end{bmatrix}$$

The axis directions of the co-rotated coordinate system can now be computed as

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} = \mathbf{R}_r \begin{bmatrix} \mathbf{r}_1^0 & \mathbf{r}_2^0 & \mathbf{r}_3^0 \end{bmatrix}$$

The position of a point on the rigidly rotated axis of the element can be obtained as

$$\mathbf{x}_R = \mathbf{x}_M + \mathbf{R}_r(\mathbf{X}_2 - \mathbf{X}_1)\left(\xi - \frac{1}{2}\right)$$

where the local coordinate ξ ranges from 0 to 1, and $\mathbf{X}_i$ denotes original node coordinates. $\mathbf{x}_M$ is the midpoint position, computed as the average of the two nodes,

$$\mathbf{x}_M = \frac{1}{2}(\mathbf{X}_1 + \mathbf{u}_1 + \mathbf{X}_2 + \mathbf{u}_2)$$

In addition to the rigid body motion described so far, there are the local deformational displacements with respect to the local rotated beam axes. The deformational displacement can be computed as the difference between the current position and the rigid body position

$$\bar{\mathbf{u}} = \mathbf{X} + \mathbf{u} - \mathbf{x}_R = \mathbf{u} - \mathbf{u}_R$$

Here, and in the following, an overline denotes a deformational quantity. The deformational rotation at the nodes is approximated by

$$\bar{\theta}_1 = -\frac{\theta_2 - \theta_1}{2}$$
$$\bar{\theta}_2 = \frac{\theta_2 - \theta_1}{2}$$

These local deformations are interpolated by the same shape functions as described in the previous section:

$$\begin{Bmatrix} \bar{\mathbf{u}} \\ \bar{\theta} \end{Bmatrix} = [\mathbf{N}] \begin{Bmatrix} \bar{\mathbf{u}}_1 \\ \bar{\theta}_1 \\ \bar{\mathbf{u}}_2 \\ \bar{\theta}_2 \end{Bmatrix}$$

The total displacement and rotation vectors can be expressed in term of the rigid motion of the local axes, followed by the deformational motion relative to these axes.

$$\mathbf{u} = \mathbf{u}_R + \bar{\mathbf{u}}$$

A deformational rotation matrix $\bar{\mathbf{R}}$ can be defined as

$$\bar{\mathbf{R}} = \mathbf{I} + \bar{\Theta}$$

where $\bar{\Theta}$ is the skew-symmetric representation of the deformational rotation vector.

The total rotation vector is computed from a total rotation matrix, $\mathbf{R}$. The total rotation matrix is first composed from the rigid body rotation and the incremental rotation.

$$\mathbf{R} = \bar{\mathbf{R}}\mathbf{R}_r$$

The total rotation vector can now be extracted from the total rotation matrix. The magnitude of the rotation vector is first computed as

$$\theta = \text{acos}\left(\frac{\text{trace}(\mathbf{R}) - 1}{2}\right)$$

The full rotation vector is then computed as

$$\theta = \frac{\theta}{2\sin\theta}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$$

To avoid singularity problems when the angle is close to zero, the gamma function is actually used in the expressions, since

$$\frac{\theta}{\sin\theta} = \gamma\left(1 + \frac{\theta}{\pi}\right)\gamma\left(1 - \frac{\theta}{\pi}\right)$$

Strain–Displacement/Rotation Relation

The axial strain depends on the rotation derivative (curvature) and axial displacement derivative defined by the shape function and the transversal coordinate in the beam. For the 3D case it becomes

$$\varepsilon = z_l \frac{\partial \theta_{yl}}{\partial s} - y_l \frac{\partial \theta_{zl}}{\partial s} + \frac{\partial u_{axi}}{\partial s}$$

where s is the local coordinate along the beam axis, and u_{axi} is the axial displacement. The coordinates from the beam centerline in the local transversal directions are denoted z_l and y_l respectively. In the 2D case, the first term is omitted, and the local z direction is always directed out of the plane.

The total strain ε consists of thermal (ε_{th}), hygroscopic (ε_{hs}), initial (ε_i), and elastic strains(ε_{el})

$$\varepsilon = \varepsilon_{el} + \varepsilon_{th} + \varepsilon_{hs} + \varepsilon_i$$

Stress–Strain Relation

The stress–strain relation for the axial deformation in the beam is described by

$$\sigma = E\varepsilon_{el} + \sigma_i$$

where E is Young's modulus, and σ_i is the initial stress.

The stress strain relation for the torsional and shear deformation is

$$\tau = G\gamma$$

where τ is the shear stress, γ is the shear strain and G is the shear modulus. Often the material data is given in terms of Young's modulus and Poisson's ratio ν, in which case G is defined as

$$G = \frac{E}{2(1+\nu)}$$

Thermal Strain

The temperature is assumed to vary linearly across the beam's cross section. For the 3D beam it becomes

$$T = T_{\mathrm{m}} + T_{\mathrm{gz}} z_{\mathrm{l}} + T_{\mathrm{gy}} y_{\mathrm{l}}$$

where T_{m} is the temperature at the beam centerline while T_{gz} and T_{gy} are the temperature gradients in the two local transversal directions. The thermal strain is thus

$$\varepsilon_{\mathrm{th}} = \alpha(T_{\mathrm{m}} + T_{\mathrm{gz}} z_{\mathrm{l}} + T_{\mathrm{gy}} y_{\mathrm{l}} - T_{\mathrm{ref}})$$

For the 2D beam, the term depending on z_{l} disappears.

Hygroscopic Swelling

The moisture concentration is assumed to vary linearly across the beam's cross section. For the 3D beam it becomes

$$c_{\mathrm{mo}} = c_{\mathrm{m}} + c_{\mathrm{gz}} z_{\mathrm{l}} + c_{\mathrm{gy}} y_{\mathrm{l}}$$

where c_{m} is the concentration at the beam centerline while c_{gz} and c_{gy} are the concentration gradients in the two local transversal directions. The strain from hygroscopic swelling is then

$$\varepsilon_{\mathrm{hs}} = \beta_{\mathrm{h}}(c_{\mathrm{m}} + c_{\mathrm{gz}} z_{\mathrm{l}} + c_{\mathrm{gy}} y_{\mathrm{l}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, and $c_{\mathrm{mo,ref}}$ is the strain-free reference moisture concentration.

For the 2D beam, the term depending on z_{l} disappears.

Initial Stress and Strain

The initial stress means the stress before any loads, displacements, and initial strains have been applied.

The initial stress distribution is given as initial forces and moments in the local section directions.

$$M_{\mathrm{iyl}} = \int_A \sigma_{\mathrm{i}} z_{\mathrm{l}} dA$$
$$M_{\mathrm{izl}} = -\int_A \sigma_{\mathrm{i}} y_{\mathrm{l}} dA$$
$$N_{\mathrm{i}} = \int_A \sigma_{\mathrm{i}} dA$$
$$M_{\mathrm{ixl}} = \int_A (\tau_{\mathrm{ixz}} y_{\mathrm{l}} - \tau_{\mathrm{iyz}} z_{\mathrm{l}}) dA$$

In 2D the x- and y-components of moments disappear.

In a Timoshenko beam, the initial shear forces appear as independent quantities, and can also be specified.

$$T_{\mathrm{iyl}} = \int_A \tau_{\mathrm{ixy}} dA$$
$$T_{\mathrm{iyl}} = \int_A \tau_{\mathrm{ixz}} dA$$

The initial strain is the strain before any loads, displacements, and initial stresses have been applied. The initial axial strain distribution is given as initial curvature and initial axial strain

$$\varepsilon_{\mathrm{i}} = z_{\mathrm{l}} \left(\frac{\partial \theta_{\mathrm{yl}}}{\partial s} \right)_{\mathrm{i}} - y_{\mathrm{l}} \left(\frac{\partial \theta_{\mathrm{zl}}}{\partial s} \right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s} \right)_{\mathrm{i}}$$

In 2D the z_{l} dependent term disappears. As initial strain for the torsional degree of freedom, the derivative of the twist angle,

$$\left(\frac{\partial \theta_{\mathrm{xl}}}{\partial s} \right)_{\mathrm{i}}$$

is used.

In a Timoshenko beam it is also possible to set the initial values for the shear strains.

Implementation

The implementation is based on the principle of virtual work, which states that the sum of virtual work from internal strains and external loads equals zero:

$$\delta W = \delta \int_V (-\varepsilon_{\mathrm{el}}\sigma + \mathbf{u}^T \mathbf{F} dV) = 0$$

The beam elements are formulated in terms of the stress resultants (normal force, bending moments and twisting moment).

The normal force is defined as

$$N = \int_A \sigma dA = \int_A (E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}}) dA =$$

$$\int_A \left\{ E\left[\left(z_{\mathrm{l}}\frac{\partial \theta_{\mathrm{yl}}}{\partial s} - y_{\mathrm{l}}\frac{\partial \theta_{\mathrm{zl}}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left[z_{\mathrm{l}}\left(\frac{\partial \theta_{\mathrm{yl}}}{\partial s}\right)_{\mathrm{i}} - y_{\mathrm{l}}\left(\frac{\partial \theta_{\mathrm{zl}}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right] - \right.\right.$$

$$\left.\left. \alpha(T_{\mathrm{m}} + T_{\mathrm{gz}} z_{\mathrm{l}} + T_{\mathrm{gy}} y_{\mathrm{l}} - T_{\mathrm{ref}}) - \beta_{\mathrm{h}}(c_{\mathrm{m}} + c_{\mathrm{gz}} z_{\mathrm{l}} + c_{\mathrm{gy}} y_{\mathrm{l}} - c_{\mathrm{mo,ref}}) \right] + \sigma_{\mathrm{i}} \right\} dA =$$

$$\int_A \left\{ E\left[\left(\frac{\partial u_{\mathrm{axi}}}{\partial s} - \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) - \alpha(T_{\mathrm{m}} - T_{\mathrm{ref}}) - \beta_{\mathrm{h}}(c_{\mathrm{m}} - c_{\mathrm{mo,ref}})\right] \right\} dA + \int_A \sigma_{\mathrm{i}} dA =$$

$$EA\left[\left(\frac{\partial u_{\mathrm{axi}}}{\partial s} - \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) - \alpha(T_{\mathrm{m}} - T_{\mathrm{ref}}) - \beta_{\mathrm{h}}(c_{\mathrm{m}} - c_{\mathrm{mo,ref}})\right] + N_{\mathrm{i}}$$

Because the local coordinates are defined with their origin at the centroid of the cross section, any surface integral of an odd power of a local coordinate evaluates to zero.

The beam bending moments are defined as

$$M_{\mathrm{yl}} = \int_A \sigma z_{\mathrm{l}} dA = \int_A z_{\mathrm{l}}(E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}})dA =$$

$$\int_A z_{\mathrm{l}}\left\{E\left[\left(z_{\mathrm{l}}\frac{\partial\theta_{\mathrm{yl}}}{\partial s} - y_{\mathrm{l}}\frac{\partial\theta_{\mathrm{zl}}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left[z_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{yl}}}{\partial s}\right)_{\mathrm{i}} - y_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{zl}}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right] - \right.\right.$$

$$\left.\left.\alpha(T_{\mathrm{m}} + T_{\mathrm{gz}}z_{\mathrm{l}} + T_{\mathrm{gy}}y_{\mathrm{l}} - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}}z_{\mathrm{l}} + c_{\mathrm{gy}}y_{\mathrm{l}} - c_{\mathrm{mo,ref}})\right] + \sigma_{\mathrm{i}}\right\}dA =$$

$$\int_A z_{\mathrm{l}}^2\left\{E\left[\frac{\partial\theta_{\mathrm{yl}}}{\partial s} - \left(\frac{\partial\theta_{\mathrm{yl}}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{gz} - \beta_{\mathrm{h}}c_{\mathrm{gz}}\right]\right\}dA + \int_A \sigma_{\mathrm{i}}z_{\mathrm{l}}dA =$$

$$EI_{\mathrm{yy}}\left[\frac{\partial\theta_{\mathrm{yl}}}{\partial s} - \left(\frac{\partial\theta_{\mathrm{yl}}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{\mathrm{gz}} - \beta_{\mathrm{h}}c_{\mathrm{gz}}\right] + M_{\mathrm{iyl}}$$

$$M_{\mathrm{zl}} = -\int_A \sigma y_{\mathrm{l}} dA = -\int_A y_{\mathrm{l}}(E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}})dA =$$

$$-\int_A y_{\mathrm{l}}\left\{E\left[\left(z_{\mathrm{l}}\frac{\partial\theta_{\mathrm{yl}}}{\partial s} - y_{\mathrm{l}}\frac{\partial\theta_{\mathrm{zl}}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left[z_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{yl}}}{\partial s}\right)_{\mathrm{i}} - y_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{zl}}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right]\right.\right.$$

$$\left.\left.- \alpha(T_{\mathrm{m}} + T_{\mathrm{gz}}z_{\mathrm{l}} + T_{\mathrm{gy}}y_{\mathrm{l}} - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}}z_{\mathrm{l}} + c_{\mathrm{gy}}y_{\mathrm{l}} - c_{\mathrm{mo,ref}})\right] + \sigma_{\mathrm{i}}\right\}dA =$$

$$-\int_A\left\{y_{\mathrm{l}}^2 \cdot \left[E\left(-\frac{\partial\theta_{\mathrm{zl}}}{\partial s} + \left(\frac{\partial\theta_{\mathrm{zl}}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{\mathrm{gy}} - \beta_{\mathrm{h}}c_{\mathrm{gy}}\right)\right] \cdot dA\right\} - \int_A \sigma_{\mathrm{i}}y_{\mathrm{l}}dA =$$

$$EI_{\mathrm{zz}}\left[\frac{\partial\theta_{\mathrm{zl}}}{\partial s} - \left(\frac{\partial\theta_{\mathrm{zl}}}{\partial s}\right)_{\mathrm{i}} + \alpha T_{\mathrm{gy}} + \beta_{\mathrm{h}}c_{\mathrm{gy}}\right] + M_{\mathrm{izl}}$$

M_{yl} is present only in 3D, and so is the torsional moment M_{xl} described below. The torsional stiffness of the beam is defined using the torsional constant J given by

$$J = \frac{M}{G\theta}l$$

In a similar way as for the bending part a torsional moment is then defined as

$$M_{\mathrm{xl}} = GJ\left[\frac{\partial\theta_{\mathrm{xl}}}{\partial s} - \left(\frac{\partial\theta_{\mathrm{xl}}}{\partial s}\right)_{\mathrm{i}}\right] + M_{\mathrm{ixl}}$$

Using the beam moment and normal force the expression for the virtual work becomes very compact:

$$\delta W = -\int_L \left[M_{yl}\delta\left(\frac{\partial \theta_{yl}}{\partial s}\right) + M_{zl}\delta\left(\frac{\partial \theta_{zl}}{\partial s}\right) + N\delta\left(\frac{\partial u_{axi}}{\partial s}\right) + M_{xl}\delta\left(\frac{\partial \theta_{xl}}{\partial s}\right) \right] dx$$

For 2D, the first and fourth terms are omitted. For the case of Timoshenko beam, there is also a shear stress contribution added,

$$\delta W = -\int_L (T_{yl}\delta\gamma_{yl} + T_{zl}\delta\gamma_{zl}) dx$$

where the second term is present only in 3D.

A special feature of some unsymmetrical cross sections is that they twist under a transversal load that is applied to beam centerline. As an example, this would be the case for a U-profile under self-weight, loaded in the stiff direction. It is only a load applied at the *shear center* which causes a pure deflection without twist. This effect can be incorporated by supplying the coordinates of the shear center in the local coordinate system (e_y, e_z). A given transversal load (f_{xl}, f_{yl}, f_{zl}), which is defined as acting along the centerline, is then augmented by a twisting moment given by

$$m_{xl} = f_{yl}e_z - f_{zl}e_y$$

Theory for Section Stiffness

The Section Stiffness node in the Beam interface allows modeling of beams with nonhomogeneous cross sections by, instead of geometrical and material properties, directly entering values for the stiffness matrix **S**.

For a 3D Euler–Bernoulli beam, the relationship between the section forces and the deformation of the beam is given by

$$\begin{bmatrix} N \\ M_{xl} \\ M_{yl} \\ M_{zl} \end{bmatrix} = \begin{bmatrix} N_i \\ M_{ixl} \\ M_{iyl} \\ M_{izl} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \frac{\partial u_{axi}}{\partial s} - \varepsilon_{ni} \\ \frac{\partial \theta_{xl}}{\partial s} - \theta_{sixl} \\ \frac{\partial \theta_{yl}}{\partial s} - \theta_{siyl} \\ \frac{\partial \theta_{zl}}{\partial s} - \theta_{sizl} \end{bmatrix} \quad (3\text{-}262)$$

including initial normal force N_{i}; moments M_{ixl}, M_{iyl}, and M_{izl}; normal strain ε_{i}; and curvatures θ_{sixl}, θ_{siyl}, and θ_{sizl}. Here, $\mathbf{S}$ is a 4-by-4 symmetric matrix. For a homogeneous prismatic beam with isotropic material properties, it is defined as

$$\mathbf{S} = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 \\ 0 & 0 & EI_{\mathrm{yy}} & 0 \\ 0 & 0 & 0 & EI_{\mathrm{zz}} \end{bmatrix}$$

For a 2D Euler–Bernoulli beam, out-of-plane moments are zero, and Equation 3-262 thus simplifies to

$$\begin{bmatrix} N \\ M_{\mathrm{zl}} \end{bmatrix} = \begin{bmatrix} N_{\mathrm{i}} \\ M_{\mathrm{izl}} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \dfrac{\partial u_{\mathrm{axi}}}{\partial s} - \varepsilon_{\mathrm{ni}} \\ \dfrac{\partial \theta_{\mathrm{zl}}}{\partial s} - \theta_{\mathrm{sizl}} \end{bmatrix}$$

with $\mathbf{S}$ now being a 2-by-2 symmetric matrix.

For a 3D Timoshenko beam, the relationship between the section forces and the deformation of the beam is given by

$$\begin{bmatrix} N \\ T_{\mathrm{yl}} \\ T_{\mathrm{zl}} \\ M_{\mathrm{xl}} \\ M_{\mathrm{yl}} \\ M_{\mathrm{zl}} \end{bmatrix} = \begin{bmatrix} N_{\mathrm{i}} \\ T_{\mathrm{iyl}} \\ T_{\mathrm{izl}} \\ M_{\mathrm{ixl}} \\ M_{\mathrm{iyl}} \\ M_{\mathrm{izl}} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \dfrac{\partial u_{\mathrm{axi}}}{\partial s} - \varepsilon_{\mathrm{ni}} \\ \gamma_{\mathrm{yl}} - \gamma_{\mathrm{iyl}} \\ \gamma_{\mathrm{zl}} - \gamma_{\mathrm{izl}} \\ \dfrac{\partial \theta_{\mathrm{xl}}}{\partial s} - \theta_{\mathrm{sixl}} \\ \dfrac{\partial \theta_{\mathrm{yl}}}{\partial s} - \theta_{\mathrm{siyl}} \\ \dfrac{\partial \theta_{\mathrm{zl}}}{\partial s} - \theta_{\mathrm{sizl}} \end{bmatrix} \tag{3-263}$$

with $\mathbf{S}$ being a 6-by-6 symmetric matrix. The equation now also includes the initial shear forces T_{iyl} and T_{izl} as well as the initial shear strains γ_{iyl} and γ_{izl}.

For a 2D Timoshenko beam, out-of-plane forces and moments are zero, and Equation 3-263 thus simplifies to

$$\begin{bmatrix} N \\ T_{\mathrm{yl}} \\ M_{\mathrm{zl}} \end{bmatrix} = \begin{bmatrix} N_{\mathrm{i}} \\ T_{\mathrm{iyl}} \\ M_{\mathrm{izl}} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \dfrac{\partial u_{\mathrm{axi}}}{\partial s} - \varepsilon_{\mathrm{ni}} \\ \gamma_{\mathrm{yl}} - \gamma_{\mathrm{iyl}} \\ \dfrac{\partial \theta_{\mathrm{zl}}}{\partial s} - \theta_{\mathrm{sizl}} \end{bmatrix}$$

with **S** now being a 3-by-3 symmetric matrix.

Using the section forces, the virtual work for the beam becomes

$$\delta W = -\int_L \left[N\delta\left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right) + T_{\mathrm{yl}}\delta\gamma_{\mathrm{yl}} + T_{\mathrm{zl}}\delta\gamma_{\mathrm{zl}} + M_{\mathrm{xl}}\delta\left(\frac{\partial \theta_{\mathrm{xl}}}{\partial s}\right) + M_{\mathrm{yl}}\delta\left(\frac{\partial \theta_{\mathrm{yl}}}{\partial s}\right) + M_{\mathrm{zl}}\delta\left(\frac{\partial \theta_{\mathrm{zl}}}{\partial s}\right)\right] dx$$

where only the relevant terms are kept, depending on the beam formulation and the space dimension.

Stress Evaluation

Since the basic result quantities for beams are the integrated stresses in terms of section forces and moment, special considerations are needed for the evaluation of actual stresses.

The normal stress from axial force is constant over the section, and computed as

$$\sigma_{\mathrm{n}} = \frac{N}{A}$$

The normal stress from bending is computed in four user-selected points $(y_{\mathrm{l}k}, z_{\mathrm{l}k})$ in the cross section as

$$\sigma_{\mathrm{b}k} = \frac{M_{\mathrm{yl}} z_{\mathrm{l}k}}{I_{\mathrm{yy}}} - \frac{M_{\mathrm{zl}} y_{\mathrm{l}k}}{I_{\mathrm{zz}}}$$

In 2D, only two points, specified by their local y-coordinates are used.

The total normal stress in these points is then

$$\sigma_k = \sigma_{\mathrm{b}k} + \sigma_{\mathrm{n}}$$

The peak normal stress in the section is defined as

$$\sigma_{max} = \max(|\sigma_k|)$$

When using the built in common cross sections, a special method is used for the Circular and Pipe sections. Since there are no extreme positions around a circle, a maximum bending stress is computed as

$$\sigma_{b,\,max} = \frac{d_o\sqrt{M_{yl}^2 + M_{zl}^2}}{2I_{zz}}$$

where d_o is the outer diameter. This stress then replaces the stress from the stress evaluation points in maximum stress expressions. This ensures that the correct peak stress is evaluated irrespective of where it appears along the circumference.

The shear stress from twist in general has a complex distribution over the cross section. The maximum shear stress due to torsion is defined as

$$\tau_{t,\,max} = \frac{|M_{xl}|}{W_t}$$

where W_t is the torsional section modulus. This result is available only in 3D.

The section shear forces are computed in two different ways depending on the beam formulation. For Euler–Bernoulli theory, the section forces proportional to the third derivative of displacement, or equivalently, the second derivative of the rotation.

$$T_{zl} = EI_{yy}\frac{\partial^2\theta_{yl}}{\partial s^2}$$

$$T_{yl} = -EI_{zz}\frac{\partial^2\theta_{zl}}{\partial s^2}$$

where T_{zl} is available only in 3D. In the case of Timoshenko theory shear force is computed directly from the shear strain.

The average shear stresses are computed from the shear forces as

$$\tau_{sz,\,ave} = \frac{T_{zl}}{A}$$
$$\tau_{sy,\,ave} = \frac{T_{yl}}{A} \qquad (3\text{-}264)$$

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\begin{aligned} \tau_{sz,\,max} &= \mu_z \tau_{sz,\,ave} \\ \tau_{sy,\,max} &= \mu_y \tau_{sy,\,ave} \end{aligned} \tag{3-265}$$

As the directions and positions of maximum shear stresses from shear and twist are not known in a general case, upper bounds to the shear-stress components are defined as

$$\begin{aligned} \tau_{xz,\,max} &= |\tau_{sz,\,max}| + \tau_{t,\,max} \\ \tau_{xy,\,max} &= |\tau_{sy,\,max}| + \tau_{t,\,max} \end{aligned}$$

The maximum von Mises equivalent stress for the cross section is then defined as

$$\sigma_{mises} = \sqrt{\sigma_{max}^2 + 3\tau_{xy,\,max}^2 + 3\tau_{xz,\,max}^2}$$

Since the maximum values for the different stress components in general occur at different positions over the cross section, the equivalent stress thus computed is a conservative approximation.

Common Cross Sections

The cross-section data for the common cross sections can be computed internally in COMSOL Multiphysics. In this section, the expressions used are summarized.

	Beam Cross Section

RECTANGULAR SECTION

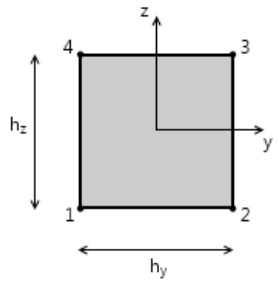


Figure 3-62: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-10: RECTANGULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$h_y h_z$	
I_{zz}	$\frac{h_y^3 h_z}{12}$	
e_z	0	
μ_y	1.5	
κ_y	5/6	
I_{yy}	$\frac{h_z^3 h_y}{12}$	
e_y	0	
μ_z	1.5	
κ_z	5/6	
J	$\frac{h_y^2 h_z^2 q}{3}\left(1-\sum_{n=1}^{2}\frac{192q}{\pi^5(2n-1)^5}\tanh\left(\frac{(2n-1)\pi}{2q}\right)\right)$	q= min(hy/hz, hz/hy)

TABLE 3-10: RECTANGULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
W_t	$\frac{J}{\sqrt{h_y h_z q}\left(1-\sum_{n=1}^{2}\frac{8}{\pi^2(2n-1)^2\cosh\left(\frac{(2n-1)\pi}{2q}\right)}\right)}$	q= min(hy/hz, hz/hy)
p_1	$\left(\frac{-h_y}{2},\frac{-h_z}{2}\right)$	
p_2	$\left(\frac{h_y}{2},\frac{-h_z}{2}\right)$	
p_3	$\left(\frac{h_y}{2},\frac{h_z}{2}\right)$	
p_4	$\left(\frac{-h_y}{2},\frac{h_z}{2}\right)$	
r_e	$\frac{h_y+h_z}{4}$	

BOX SECTION

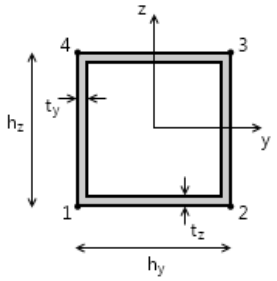


Figure 3-63: Geometry of a box shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-11: BOX SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$2(h_y t_z + h_z t_y) - 4 t_y t_z$	
I_{zz}	$\frac{t_z h_y^3 + t_y^3 (h_z - 2t_z)}{6} + \frac{t_y (h_z - 2t_z)(h_y - t_y)^2}{2}$	
e_z	0	
μ_y	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - 2t_z))A}{16 t_z I_{zz}}$	
κ_y	$\frac{2 h_y t_z}{A}$	
I_{yy}	$\frac{t_y h_z^3 + t_z^3 (h_y - 2t_y)}{6} + \frac{t_z (h_y - 2t_y)(h_z - t_z)^2}{2}$	
e_y	0	
μ_z	$\frac{(h_z^2 h_y - (h_z - 2t_z)^2 (h_y - 2t_y))A}{16 t_y I_{yy}}$	
κ_z	$\frac{2 h_z t_y}{A}$	
J	$\frac{2(h_y - t_y)^2 (h_z - t_z)^2}{\frac{h_y - t_y}{t_z} + \frac{h_z - t_z}{t_y}}$	Thin-walled approximation
W_t	$2(h_y - t_y)(h_z - t_z) \cdot \min(t_y, t_z)$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, \frac{-h_z}{2}\right)$	
p_2	$\left(\frac{h_y}{2}, \frac{-h_z}{2}\right)$	
p_3	$\left(\frac{h_y}{2}, \frac{h_z}{2}\right)$	

TABLE 3-11: BOX SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
p_4	$\left(\frac{-h_y}{2}, \frac{h_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

CIRCULAR SECTION

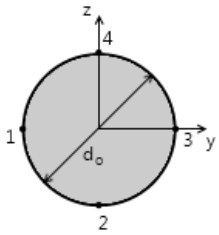


Figure 3-64: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

C

TABLE 3-12: CIRCULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$\frac{\pi d_o^2}{4}$	
I_{zz}	$\frac{\pi d_o^4}{64}$	
e_z	0	
μ_y	$\frac{4}{3}$	
κ_y	0.9	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	

TABLE 3-12: CIRCULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
κ_z	κ_y	
J	$\frac{\pi d_o^4}{32}$	
W_t	$\frac{\pi d_o^3}{16}$	
p_1	$\left(\frac{-d_o}{2}, 0\right)$	
p_2	$\left(0, \frac{-d_o}{2}\right)$	
p_3	$\left(\frac{d_o}{2}, 0\right)$	
p_4	$\left(0, \frac{d_o}{2}\right)$	
r_e	$\frac{d_o}{2}$	

PIPE SECTION

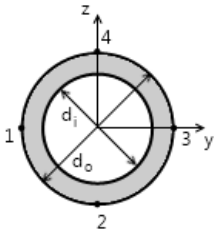


Figure 3-65: Geometry of a pipe cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-13: PIPE SECTION CONSTANTS.

PROPERTY	FORMULA	REMARKS
A	$\frac{\pi(d_o^2 - d_i^2)}{4}$	
I_{zz}	$\frac{\pi(d_o^4 - d_i^4)}{64}$	
e_z	0	
μ_y	$\frac{(d_o^3 - d_i^3)A}{12(d_o - d_i)I_{zz}}$	
κ_y	$1 + \frac{d_i}{d_o}$	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	
κ_z	κ_y	
J	$\frac{\pi(d_o^4 - d_i^4)}{32}$	
W_t	$\frac{\pi(d_o^4 - d_i^4)}{16 d_o}$	
p_1	$\left(\frac{-d_o}{2}, 0\right)$	
p_2	$\left(0, \frac{-d_o}{2}\right)$	
p_3	$\left(\frac{d_o}{2}, 0\right)$	

TABLE 3-13: PIPE SECTION CONSTANTS.

PROPERTY	FORMULA	REMARKS
p_4	$\left(0, \frac{d_o}{2}\right)$	
r_e	$\frac{d_o}{2}$	

H-PROFILE SECTION

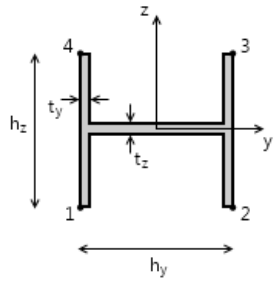


Figure 3-66: Geometry of an H-profile cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-14: H-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$2h_z t_y + t_z(h_y - 2t_y)$	
I_{zz}	$\frac{2h_z t_y^3 + t_z(h_y - 2t_y)^3}{12} + \frac{t_y h_z (h_y - t_y)^2}{2}$	
e_z	0	
μ_y	$\frac{[4h_z t_y(h_y - t_y) + t_z(h_y - 2t_y)^2]A}{8t_z I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{2t_y h_z^3 + t_z^3(h_y - 2t_y)}{12}$	

TABLE 3-14: H-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
e_y	0	
μ_z	$\frac{(h_z^2 - t_z^2)A}{8I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{2t_y^3 h_z + t_z^3(h_y - 2t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, \frac{-h_z}{2}\right)$	
p_2	$\left(\frac{h_y}{2}, \frac{-h_z}{2}\right)$	
p_3	$\left(\frac{h_y}{2}, \frac{h_z}{2}\right)$	
p_4	$\left(\frac{-h_y}{2}, \frac{h_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

U-PROFILE SECTION

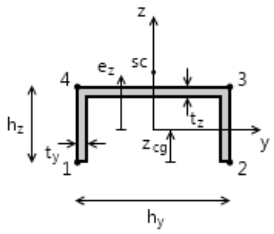


Figure 3-67: Geometry of a U-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-15: U-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$h_y t_z + 2(h_z - t_z)t_y$	
I_{zz}	$\frac{t_z h_y^3 + 2t_y^3(h_z - t_z)}{12} + \frac{t_y(h_z - t_z)(h_y - t_y)^2}{2}$	
z_{CG}	$\frac{h_y t_z(2h_z - t_z) + 2t_y(h_z - t_z)^2}{2A}$	
e_z	$h_z - \frac{t_z}{2} + \frac{t_y(2h_z - t_z)^2(h_y - t_y)^2}{16 I_{zz}} - z_{CG}$	
μ_y	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2(h_z - t_z))A}{8 t_z I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{8t_y h_z^3 + t_z^3(h_y - 2t_y) + 3t_z(h_y - 2t_y)(2h_z - t_z)^2}{12}$ $-z_{CG}{}^2 A$	
e_y	0	

TABLE 3-15: U-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
μ_z	$\frac{z_{CG}{}^2 A}{2I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{2t_y^3 h_z + t_z^3(h_y - 2t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, -z_{CG}\right)$	
p_2	$\left(\frac{h_y}{2}, -z_{CG}\right)$	
p_3	$\left(\frac{h_y}{2}, h_z - z_{CG}\right)$	
p_4	$\left(\frac{-h_y}{2}, h_z - z_{CG}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

T-PROFILE SECTION

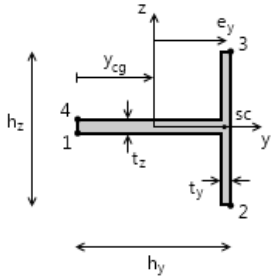


Figure 3-68: Geometry of a T-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-16: T-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$h_z t_y + (h_y - t_y) t_z$	
y_{CG}	$\frac{t_z (h_y - t_y)^2 + t_y h_z (2h_y - t_y)}{2A}$	
I_{zz}	$\frac{4t_z (h_y - t_y)^3 + h_z t_y^3 + 3t_y h_z (2h_y - t_y)^2}{12} - y_{CG}{}^2 A$	
e_z	0	
μ_y	$\frac{y_{CG}{}^2 A}{2I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{t_y h_z^3 + t_z^3 (h_y - t_y)}{12}$	
e_y	$h_y - \frac{t_y}{2} - y_{CG}$	
μ_z	$\frac{(h_z^2 - t_z^2) A}{8 I_{yy}}$	
κ_z	$\frac{4 h_z t_y}{3A}$	
J	$\frac{t_y^3 h_z + t_z^3 (h_y - t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
p_1	$\left(-y_{CG}, \frac{-t_z}{2}\right)$	
p_2	$\left(h_y - y_{CG}, \frac{-h_z}{2}\right)$	

TABLE 3-16: T-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
p_3	$\left(h_y - y_{CG}, \frac{h_z}{2}\right)$	
p_3	$\left(-y_{CG}, \frac{t_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

C-PROFILE SECTION

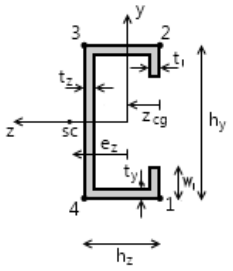


Figure 3-69: Geometry of a C-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-17: C-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$h_y t_z + 2(h_z - t_z)t_y + 2(w_1 - t_y)t_1$	q
I_{zz}	$\frac{t_z h_y^3 + 2t_y^3(h_z - t_z) + 2(w_1 - t_y)t_1^3}{12} + \frac{t_y(h_z - t_z)(h_y - t_y)^2 + t_1(w_1 - t_y)(h_y - t_y - w_1)^2}{2}$	q
z_{CG}	$\frac{h_y t_z(2h_z - t_z) + 2t_y(h_z - t_z)^2 + 2(w_1 - t_y)t_1^2}{2A}$	q

TABLE 3-17: C-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
e_z		See below the table
μ_y	$[h_y^2 t_z + 4(h_z - t_z)t_y(h_y - t_y) + 4(w_1 - t_y)t_1(h_y - t_y - w_1)]\frac{A}{8t_z I_{zz}}$	q
κ_y	$\frac{(h_y - 2t_y)t_z}{A}$	q
I_{yy}	$\frac{t_z^3 h_y + 8(h_z - t_z)^3 t_y + 8t_1^3(w_1 - t_y)}{12} + \frac{3t_z h_y (2h_z - t_z)^2}{12} - z_{CG}{}^2 A$	
e_y	0	
μ_z	$\frac{(t_y z_{CG}{}^2 + (w_1 - t_y)t_1(2z_{CG} - t_1))A}{2t_y I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{t_z^3 h_y + 2t_y^3(h_z - t_z) + 2t_1^3(w_1 - t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z, t_1)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, -z_{CG}\right)$	
p_2	$\left(\frac{h_y}{2}, -z_{CG}\right)$	
p_3	$\left(\frac{h_y}{2}, h_z - z_{CG}\right)$	

TABLE 3-17: C-PROFILE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
p_4	$\left(\frac{-h_y}{2}, h_z - z_{CG}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

The expression for the shear center offset is valid only for the case when the thickness is constant, that is

$$t_y = t_z = t_l = t$$

Then

$$e_z = h_z - \frac{t_z}{2} + \frac{N}{D} - z_{CG}$$

where

$$N = 2(h_z - t)(6(h_y - t)^2(h_z - t) + 6(h_y - t)^2(2w_l - t) - 2(2w_l - t)^3)$$
$$D = 4((h_y - t)^3 + 6(h_y - t)^2(h_z - t) + 3((h_y - t)^2(2w_l - t)) + (2w_l - t)^3) +$$
$$-12(h_y - t)(h_z - t)^2$$

HAT SECTION

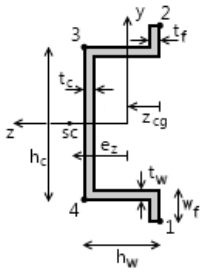


Figure 3-70: Geometry of a Hat cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-18: HAT SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$h_c t_c + 2(h_w - t_c)t_w + 2(w_f - t_w)t_f$	
I_{zz}	$\frac{t_c h_c^3 + 2t_w^3(h_w - t_c) + 2(w_f - t_w)t_f^3}{12} + \frac{t_w(h_w - t_c)(h_c - t_w)^2 + t_f(w_f - t_w)(h_c + w_f)^2}{2}$	
z_{CG}	$\frac{h_c t_c(2h_w - t_c) + 2t_w(h_w - t_c)^2 + 2(w_f - t_w)t_f^2}{2A}$	
e_z		See below the table
μ_y	$[h_c^2 t_c + 4(h_w - t_c)t_w(h_c - t_w) + 4(w_f - t_w)t_f(h_c + w_f)]\frac{A}{8t_c I_{zz}}$	
κ_y	$\frac{(h_c - 2t_w)t_c}{A}$	
I_{yy}	$\frac{t_c^3 h_c + 8(h_w - t_c)^3 t_w + 8t_f^3(w_f - t_w)}{12} + \frac{3t_c h_c(2h_w - t_c)^2}{12} - z_{CG}^2 A$	
e_y	0	
μ_z	$\frac{(t_w z_{CG}^2 + (w_f - t_w)t_f(2z_{CG} - t_f))A}{2t_w I_{yy}}$	
κ_z	$\frac{5h_w t_w}{3A}$	
J	$\frac{t_c^3 h_c + 2t_w^3(h_w - t_c) + 2t_f^3(w_f - t_w)}{3}$	Thin-walled approximation

TABLE 3-18: HAT SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
W_t	$\frac{J}{\max(t_y, t_z, t_f)}$	Thin-walled approximation
p_1	$\left(\frac{-(h_c + w_f - t_w)}{2}, -z_{CG}\right)$	
p_2	$\left(\frac{(h_c + w_f - t_w)}{2}, -z_{CG}\right)$	
p_3	$\left(\frac{h_y}{2}, h_z - z_{CG}\right)$	
p_4	$\left(\frac{-h_y}{2}, h_z - z_{CG}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

The expression for the shear center offset is valid only for the case when the thickness is constant, that is

$$t_c = t_w = t_f = t$$

Then

$$e_z = h_w - \frac{t_z}{2} + \frac{N}{D} - z_{CG}$$

where

$$N = 2(h_w - t)(6(h_c - t)^2(h_w - t) + 6(h_c - t)^2(2w_f - t) - 2(2w_f - t)^3)$$

$$D = 4((h_c - t)^3 + 6(h_c - t)^2(h_w - t) + 3((h_c - t)^2(2w_f - t)) + (2w_f - t)^3) + 12(h_c - t)(h_w - t)^2$$

Theory for the Beam Cross Section Interface

In this section the theory for the Beam Cross Section interface is described:

- Cross-Section Properties
- Computation of Stresses

Cross-Section Properties

The following cross-section properties computed by the Beam Cross Section interface are described in this section:

- Area
- Center of Gravity
- Moments of Inertia
- Directions of Principal Axes
- Local Coordinates
- Bending Shear Stresses
- Torsion
- Warping

AREA

The area is computed as

$$A = \int_A dA$$

CENTER OF GRAVITY

The center of gravity is computed as

$$x_{\mathrm{CG}} = \frac{1}{A}\int_A x\,dA$$

$$y_{\mathrm{CG}} = \frac{1}{A}\int_A y\,dA$$

MOMENTS OF INERTIA

The moments of inertia in the *XY* coordinate system are

$$I_{xx} = \int_A (y - y_{CG})^2 dA$$

$$I_{yy} = \int_A (x - x_{CG})^2 dA$$

$$I_{xy} = \int_A (x - x_{CG})(y - y_{CG}) dA$$

Since the input data required by the Beam interface is the principal moments of inertia, these must also be computed. Using the radius of the Mohr's circle:

$$R = \sqrt{\left(\frac{I_{xx} - I_{yy}}{2}\right)^2 + I_{xy}^2}$$

the principal moments of inertia can be expressed as

$$I_1 = \frac{I_{xx} + I_{yy}}{2} + R$$

$$I_2 = \frac{I_{xx} + I_{yy}}{2} - R$$

As an auxiliary variable, the radius of gyration is computed, using the expression

$$r_g = \sqrt{\frac{I_1 + I_2}{A}}$$

DIRECTIONS OF PRINCIPAL AXES

The angle needed to rotate the x-axis to the axis of the largest principal moment of inertia (x_1) is denoted α. From the definition of Mohr's circle, the angle is

$$\alpha = \frac{1}{2}\text{atan}\left(\frac{-2I_{xy}}{I_{xx} - I_{yy}}\right)$$

When implemented using the `atan2` function, the angle can be correctly evaluated for all rotations, and returns in the interval $-\pi < \alpha < \pi$.

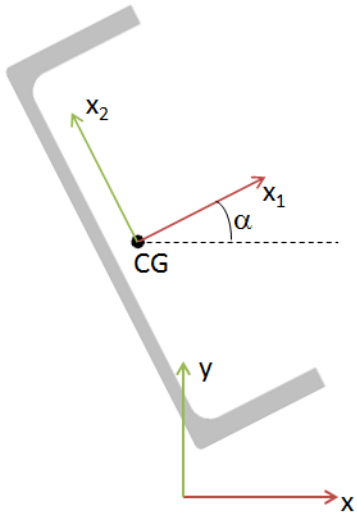


Figure 3-71: Local coordinate system and rotation angle.

LOCAL COORDINATES

The local coordinate system, having its origin in the center of gravity, and orientation given by the principal moments of inertia is given by

$$x_1 = (x - x_{CG})\cos\alpha + (y - y_{CG})\sin\alpha$$
$$x_2 = (y - y_{CG})\cos\alpha - (x - x_{CG})\sin\alpha$$

BENDING SHEAR STRESSES

The shear stresses caused by bending cannot be given a simple closed form solution, but must be solved using two independent partial differential equations, one for the force in each direction. The complete derivation is given at the end of this section. First some quantities computed from the shear stresses are defined.

The following notation is used: $\tau_{2z}^{(1)}$. This is a shear stress in the x_2 direction (acting on the plane with z as normal) caused by a unit shear force acting in the x_1 direction.

Max Shear Stress Factor

The *max shear stress factor* is the ratio between the maximum shear stress in the cross section and the average shear stress. For a shear force in the x_1 direction, the definition is

$$\mu_1 = \frac{\max(\tau^{(1)})}{\frac{1}{A}} = \max(\tau^{(1)}) \cdot A$$

where

$$\tau^{(1)} = \sqrt{(\tau_{1z}^{(1)})^2 + (\tau_{2z}^{(1)})^2}$$

is the resulting shear stress from a unit load in the x_1 direction. Similarly:

$$\mu_2 = \max(\tau^{(2)}) \cdot A$$

Shear Correction Factor

The *shear correction factor* is also computed. The shear correction factor is a multiplier which makes the strain energy from the average shear stress and shear strain in the cross section equal to the true shear energy in the cross section. The shear correction factor can be introduced through the concept *shear area*. The shear area is the reduced area which should replace the true area when computing the shear deformation of a beam. In terms of the shear correction factor it can be written as

$$A_s^{(1)} = \kappa_1 A$$

where κ_1 is the shear correction factor for a shear force in the x_1 direction. Thus, for a shear flexible beam, the constitutive relation for the average shear is

$$\bar{\gamma} = \frac{\bar{\tau}}{\kappa G} = \frac{T}{\kappa G A} = \frac{T}{G A_s}$$

To compute the shear correction factor, the true strain energy based on the actual stress and strain distribution is set equal to the strain energy from the average shear stress, when acting over the shear area. The full energy expression is

$$\frac{1}{2}\int_A (\tau_{1z}^{(1)}\gamma_{1z}^{(1)} + \tau_{2z}^{(1)}\gamma_{2z}^{(1)})\, dA$$

The strain energy based on the averaged shear stress and shear strain is

$$\frac{1}{2}\overline{\tau_{1z}^{(1)}}\,\overline{\gamma_{1z}^{(1)}}A = \frac{1}{2}\frac{T_1}{A}\overline{\gamma_{1z}^{(1)}}A = \frac{1}{2\kappa_1 G}\left(\frac{T_1}{A}\right)^2 A$$

giving

$$\frac{1}{2G}\int_A ((\tau_{1z}^{(1)})^2 + (\tau_{2z}^{(1)})^2)\,dA = \frac{T_1^2}{2G\kappa_1 A}$$

Since T_1 is a unit shear force, the shear correction factor can be computed as

$$\kappa_1 = \left(A\int_A (\tau^{(1)})^2 dA\right)^{-1}$$

Similarly:

$$\kappa_2 = \left(A\int_A (\tau^{(2)})^2 dA\right)^{-1}$$

Shear Center Distance

The *shear center* (or, equivalently, the *center of rotation*) is the point around which the shear stresses from bending has no torque. In COMSOL Multiphysics it is represented as the distance from the center of gravity of the cross section in the principal axes coordinate system. The torque can be computed as

$$0 = \int_A (-\tau_{1z}(e_2 - x_2) + \tau_{2z}(e_1 - x_1))\,dA =$$
$$\int_A (\tau_{1z}x_2 + -\tau_{2z}x_1)\,dA + e_1\int_A \tau_{2z}\,dA - e_2\int_A \tau_{1z}\,dA$$

Since there are two separate solutions for the shear stresses, it is possible to split the determination of the two shear center coordinates into

$$e_1 = -\int_A (\tau_{1z}^{(2)}x_2 + -\tau_{2z}^{(2)}x_1)\,dA$$
$$e_2 = \int_A (\tau_{1z}^{(1)}x_2 + -\tau_{2z}^{(1)}x_1)\,dA$$

Here the fact that the shear force resultant has a unit value has been used.

Derivation of the Equations for Computing the Shear Stresses

Basic beam theory assumptions give the following stress components:

$$\begin{aligned}\sigma_1 &= 0\\ \sigma_2 &= 0\\ \sigma_z &= \frac{M_1 x_2}{I_1} - \frac{M_2 x_1}{I_2}\\ \tau_{12} &= 0\end{aligned} \tag{3-266}$$

The shear forces are related to the bending moments through

$$\begin{aligned}T_1 &= -\frac{\partial M_2}{\partial z}\\ T_2 &= \frac{\partial M_1}{\partial z}\end{aligned}$$

The static equilibrium equations are

$$\begin{aligned}\sigma_{1,1} + \tau_{12,2} + \tau_{1z,z} &= 0\\ \tau_{12,1} + \sigma_{2,2} + \tau_{2z,z} &= 0\\ \tau_{1z,1} + \tau_{2z,2} + \sigma_{z,z} &= 0\end{aligned}$$

Insertion of the known stresses into the equilibrium equations gives

$$\begin{aligned}\tau_{1z,z} &= 0\\ \tau_{2z,z} &= 0\\ \tau_{1z,1} + \tau_{2z,2} + \frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2} &= 0\end{aligned}$$

The two first equations simply state that the shear stresses are independent of z, whereas the third equation is the one on which to focus the interest. Assume that the shear stresses can be derived from a scalar stress potential, ψ, through:

$$\begin{aligned}\tau_{1z} &= \psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu) I_2}\\ \tau_{2z} &= \psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu) I_1}\end{aligned}$$

Insertion of this assumption into the third equilibrium equation gives the Poisson type equation

$$\Delta\psi = -\left(\frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2}\right) \tag{3-267}$$

In addition to equilibrium, also compatibility must be fulfilled. The Beltrami–Michell form of compatibility equations includes the assumption of an isotropic linear elastic material, and then states that

$$\Delta\sigma_{ij} + \frac{1}{1+\nu}\frac{\partial(\text{trace}(\sigma))}{\partial x_i \partial x_j} = 0$$

Given the stress state from Equation 3-266, the only two nontrivial equations are

$$\Delta\tau_{1z} + \frac{1}{1+\nu}\frac{T_1}{I_2} = 0$$
$$\Delta\tau_{2z} + \frac{1}{1+\nu}\frac{T_2}{I_1} = 0$$

Inserting the assumed stress components gives

$$\Delta\psi_{,1} + \frac{\nu T_1}{(1+\nu)I_2} + \frac{1}{1+\nu}\frac{T_1}{I_2} = \Delta\psi_{,1} + \frac{T_1}{I_2} = 0$$
$$\Delta\psi_{,2} + \frac{\nu T_2}{(1+\nu)I_1} + \frac{1}{1+\nu}\frac{T_2}{I_1} = \Delta\psi_{,2} + \frac{T_2}{I_1} = 0$$

Integration of the first equation with respect to x_1 and the second equation with respect to x_2 gives

$$\Delta\psi + \frac{T_1 x_1}{I_2} + f_2(x_2) = 0$$
$$\Delta\psi + \frac{T_2 x_2}{I_1} + f_1(x_1) = 0$$

Combining these two equations results in

$$\Delta\psi = -\left(\frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2}\right)$$

This is the same equation as Equation 3-267. It is thus possible to fulfill equilibrium, compatibility, and the constitutive relation with a single equation of Poisson type.

On all free boundaries, the stress must be zero, giving the condition

$$\tau_{1z} n_1 + \tau_{2z} n_2 = 0$$

Using the assumed shear stresses, this can be converted into the Neumann condition

$$\nabla\psi \cdot \mathbf{n} = -\left(\frac{\nu T_1 x_2^2}{2(1+\nu) I_2} n_1 + \frac{\nu T_2 x_1^2}{2(1+\nu) I_1} n_2 \right)$$

It must also be determined that the resultant of the shear stresses actually match the applied shear forces, that is

$$\int_A \tau_{1z}\, dA = T_1$$

$$\int_A \tau_{2z}\, dA = T_2$$

The proofs for the two components are analogous, so it is shown only for the x_1 direction:

$$\int_A \tau_{1z}\, dA = \int_A \left(\psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu) I_2} \right) dA = \int_A \psi_{,1} dA + \frac{\nu T_1 I_1}{2(1+\nu) I_2}$$

To compute the integral of the x_1-derivative of ψ, a term containing the differential equation itself is added. This is a zero contribution, but it makes further simplifications possible.

$$\int_A \psi_{,1} dA = \int_A \left(\psi_{,1} + x_1 \left(\Delta\psi + \frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2} \right) \right) dA =$$

$$\int_A (\psi_{,1} + x_1 \Delta\psi)\, dA + T_1 = \int_A \left(\frac{\partial}{\partial x_1}(x_1 \psi_{,1}) + \frac{\partial}{\partial x_2}(x_1 \psi_{,2}) \right) dA + T_1 =$$

$$\oint_\Gamma ((x_1 \psi_{,1}) n_1 + (x_1 \psi_{,2}) n_2)\, ds + T_1 = \oint_\Gamma x_1 (\nabla\psi \cdot \mathbf{n}) ds + T_1 =$$

$$\oint_\Gamma x_1 \left(-\left(\frac{\nu T_1 x_2^2}{2(1+\nu) I_2} n_1 + \frac{\nu T_2 x_1^2}{2(1+\nu) I_1} n_2 \right) \right) ds + T_1 =$$

$$\int_A \left(-\frac{\nu T_1 x_2^2}{2(1+\nu) I_2} \right) dA + T_1 = \frac{-\nu T_1 I_1}{2(1+\nu) I_2} + T_1$$

In the transformations above these facts are used:

- The area integral of x_1, x_1, or $x_1 x_2$ are zero since the coordinate system is positioned at the center of gravity and is oriented along the principal axes.
- The area integral of x_2^2 is I_1.
- The divergence theorem is used for transformation between area and surface integrals.
- The Neumann condition on the boundary is inserted into the integral along the boundary.

This proves that the assumed stress field also produces the correct resultants.

When internal holes are present, it is necessary ensure compatibility in the sense that the displacement is single valued when going around the hole:

$$\oint_\Gamma du_1 = \oint_\Gamma du_2 = \oint_\Gamma du_z = 0$$

The displacements can be derived from the strains, which are given by the stress state

$$\varepsilon_1 = \frac{\partial u_1}{\partial x_1} = -\nu\varepsilon_z$$

$$\varepsilon_2 = \frac{\partial u_2}{\partial x_2} = -\nu\varepsilon_z$$

$$\varepsilon_z = \frac{\partial u_z}{\partial z} = \frac{M_1 x_2}{EI_1} - \frac{M_2 x_1}{EI_2}$$

$$\gamma_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} = 0$$

$$\gamma_{1z} = \frac{\partial u_1}{\partial z} + \frac{\partial u_z}{\partial x_1} = \frac{1}{G}\left(\psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2}\right)$$

$$\gamma_{2z} = \frac{\partial u_2}{\partial z} + \frac{\partial u_z}{\partial x_2} = \frac{1}{G}\left(\psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1}\right)$$

Integration of the direct strains gives

$$u_1 = -\frac{\nu M_1 x_1 x_2}{EI_1} + \frac{\nu M_2 x_1^2}{2EI_2} + g_1(x_2, z)$$

$$u_2 = -\frac{\nu M_1 x_2^2}{2EI_1} + \frac{\nu M_2 x_1 x_2}{EI_2} + g_2(x_1, z)$$

Since the only part of the displacement that is relevant for the bending shear stresses is independent of the z-coordinate, the functions g_1 and g_2 can be considered as independent of z:

$$\oint_\Gamma du_1 = \oint_\Gamma \left(\frac{\partial u_1}{\partial x_1} dx_1 + \frac{\partial u_1}{\partial x_2} dx_2\right) =$$

$$\oint_\Gamma \left(\left(-\left(\frac{\nu M_1 x_2}{EI_1} - \frac{\nu M_2 x_1}{EI_2}\right)\right) dx_1 + \left(-\frac{\nu M_1 x_1}{EI_1} + \frac{\partial g_1}{\partial x_2}\right) dx_2\right) =$$

$$\int_A \left(\frac{\nu M_1 x_1 x_2}{EI_1}\right) dA - \int_A \left(\frac{\nu M_1 x_1 x_2}{EI_1}\right) dA = 0$$

In the last transformation Green's theorem is used. The uniqueness of the u_2 displacement can be shown in the same way.

The uniqueness of the out-of-plane component of the displacement is shown in Equation 3-268:

$$\begin{aligned}
&\oint_\Gamma du_z = \oint_\Gamma \left(\frac{\partial u_z}{\partial x_1}dx_1 + \frac{\partial u_z}{\partial x_2}dx_2\right) = \oint_\Gamma \left(\left(\gamma_{1z} - \frac{\partial u_1}{\partial z}\right)dx_1 + \left(\gamma_{2z} - \frac{\partial u_2}{\partial z}\right)dx_2\right) = \\
&\oint_\Gamma \left(\frac{1}{G}\left(\psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2}\right) - \left(\frac{\nu T_1 x_1 x_2}{EI_1} + \frac{\nu T_2 x_1^2}{2EI_2}\right)\right)dx_1 + \\
&\oint_\Gamma \left(\frac{1}{G}\left(\psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1}\right) - \left(\frac{\nu T_1 x_2^2}{2EI_1} + \frac{\nu T_2 x_1 x_2}{EI_2}\right)\right)dx_2 = \\
&-\int_A \left(\frac{1}{G}\left(\psi_{,12} + \frac{\nu T_1 x_2}{(1+\nu)I_2}\right) - \frac{\nu T_1 x_1}{EI_1}\right)dA + \\
&\int_A \left(\frac{1}{G}\left(\psi_{,21} + \frac{\nu T_2 x_1}{(1+\nu)I_1}\right) - \frac{\nu T_2 x_2}{EI_2}\right)dA = 0
\end{aligned} \tag{3-268}$$

In the last step of Equation 3-268, all integrals are zero since the coordinate system is located at the center of gravity of the section. This proves that all displacement components are unique.

When solving the problem, the shear stresses caused by a unit force in each of the two principal directions must be separated, so two separate problems are solved. For the force in the x_1 direction it is formulated as

$$\Delta\psi^{(1)} = -\frac{x_1}{I_2}$$

with

$$\nabla\psi^{(1)} \cdot \mathbf{n} = \frac{-\nu x_2^2}{2(1+\nu)I_2}n_1$$

on all boundaries. The stresses are computed as

$$\begin{aligned}
\tau_{1z}^{(1)} &= \psi_{,1}^{(1)} + \frac{\nu x_2^2}{2(1+\nu)I_2} \\
\tau_{2z}^{(1)} &= \psi_{,2}^{(1)}
\end{aligned}$$

The corresponding problem for the x_2 direction is

$$\Delta\psi^{(2)} = -\frac{x_2}{I_1}$$

$$\nabla\psi^{(2)} \cdot \mathbf{n} = \frac{-\nu x_1^2}{2(1+\nu)I_1} n_1$$

$$\tau_{1z}^{(2)} = \psi_{,1}^{(2)}$$
$$\tau_{2z}^{(2)} = \psi_{,2}^{(2)} + \frac{\nu x_1^2}{2(1+\nu)I_1}$$

TORSION

The torsional properties cannot in general be computed using a closed form expression. Determining the torsional rigidity requires the solution of a PDE over the cross section. There are two ways to do this: Using a warping function or using the Prandtl stress function. The Prandtl stress function approach is used in COMSOL Multiphysics since it gives easier boundary conditions.

The general torsion theory includes the shear modulus and angle of twist, but these properties are not needed to determine the torsional rigidity, so both parameters are treated as having the value 1. In that case the equation to be solved can be simplified to

$$\Delta\phi = -2$$

where ϕ is the stress function. For a singly connected region the boundary condition is $\phi = 0$ along the whole boundary. Having solved this problem, the torsional rigidity can be computed as

$$J = 2\int_A \phi \, dA$$

The shear stresses are defined as

$$\tau_{xz} = \frac{\partial\phi}{\partial y}$$
$$\tau_{yz} = -\frac{\partial\phi}{\partial x}$$

The torsional modulus can be determined as

$$W_t = \frac{J}{\max(|\nabla\phi|)}$$

In the case that there are internal holes in the section the situation is slightly more complex. The $\phi = 0$ condition is now applicable only to the exterior boundary, whereas each boundary of an internal hole i needs a Dirichlet boundary condition:

$$\phi = H_i$$

where H_i is a constant to be determined. The constant value of the stress function fulfills the stress-free boundary conditions. There is also a compatible condition that must be fulfilled: the displacements must be single valued when going around each hole along its boundary Γ_i. This is trivially fulfilled for the in-plane displacements, but the out-of-plane displacement, w, generates the necessary equations to determine H_i:

$$0 = \oint_{\Gamma_i} dw = \oint_{\Gamma_i} \left[\frac{\partial w}{\partial x}dx + \frac{\partial w}{\partial y}dy\right] = \oint_{\Gamma_i} \left[\left(\gamma_{xz} - \left(\frac{\partial u}{\partial z}\right)\right)dx + \left(\gamma_{yz} - \left(\frac{\partial v}{\partial z}\right)\right)dy\right]$$
$$= \oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y\right)dx + \left(-\frac{\partial \phi}{\partial x} - x\right)dy\right] = \oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y\right)t_x - \left(\frac{\partial \phi}{\partial x} + x\right)t_y\right]ds$$

Here, it has been used so that the strains are equal to the stresses since the shear modulus is set to 1. The kinematic assumption that the in-plane displacements can be written as

$$u = -yz$$
$$v = xz$$

is employed. This assumption implies that the origin of the coordinate system is at the center of rotation. This is true only for doubly symmetric sections, but adding a constant offset to the x and y coordinates does not contribute to the integral.

The gradient of ϕ depends linearly on the yet unknown variables H_i, the values of which can be solved by adding one equation,

$$\oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y\right)t_x - \left(\frac{\partial \phi}{\partial x} + x\right)t_y\right]ds = 0$$

for each hole.

The expression for the torsional rigidity must in this case be augmented to

$$J = 2\int_A \phi\, dA + 2\sum_i H_i A_i$$

where A_i is the area of hole i.

WARPING

The warping properties of the cross section are not used by the Beam interface in COMSOL Multiphysics, since an assumption of pure Saint-Venant torsion is used. The data can still be useful to do manual estimates.

The warping function $\omega(x, y)$ describes the out-of-plane deformation related to torsion. It fulfills Laplace's equation $\Delta\omega = 0$.

The boundary conditions giving stress-free boundaries:

$$\nabla\omega \cdot \mathbf{n} = (x - e_x)t_x + (y - e_y)t_y$$

The offset by the shear center coordinates (e_x, e_y) is introduced because the torsion theory assumes that the coordinate system has its origin in the center of rotation (which is the same as the shear center).

The level of the warping function must also be fixed by adding a Dirichlet condition in a point. The actual value is however difficult to set. Instead it is easier to solve for a shifted warping function

$$\omega_s(x,y) = \omega(x,y) + C$$

The shifted warping function can be set to zero in an arbitrary point. The true warping function is then computed as

$$\omega(x,y) = \omega_s(x,y) - \frac{1}{A}\int_A \omega_s(x,y)\, dA$$

This criterion expresses that the average of the warping function must be zero since the axial stresses induced by torsion should not have a resultant.

The warping displacement, u, for a beam with unrestrained warping can be computed by providing the twist of the beam, ϕ_{xl} as input:

$$u(x,y) = \phi_{xl}\omega(x,y)$$

The warping constant, which is used in analysis of nonuniform torsion, is defined as

$$C_w = \int_A \omega(x,y)^2 dA$$

The axial stress caused by nonuniform torsion is

$$\sigma_w(x,y) = \frac{B\,\omega(x,y)}{C_w}$$

where B is the *bimoment*. The maximum axial stress is

$$\sigma_{w,\max} = \frac{|B|\max(|\omega(x,y)|)}{C_w} = \frac{|B|}{K_w}$$

The warping modulus is then defined as

$$K_w = \frac{\max(|\omega(x,y)|)}{C_w}$$

Given the warping constant, it is possible to compute a nondimensional number that can be used to characterize the influence of nonuniform torsion in a beam with a certain length L. This number is

$$\frac{\pi^2 E C_w}{GJL^2} = \frac{2(1+\nu)\pi^2 C_w}{JL^2}$$

Since the length is independent of the cross section, the sensitivity number is defined as

$$s_w = \frac{2(1+\nu)\pi^2 C_w}{J}$$

It has the physical dimension length squared.

Computation of Stresses

The stresses are computed using the following expressions:

- Axial Stress
- Bending Axial Stresses
- Bending Shear Stresses
- Torsional Shear Stresses
- Equivalent Stress

AXIAL STRESS

The axial stress is computed as

$$\sigma_N = \frac{N}{A}$$

where N is the axial force.

BENDING AXIAL STRESSES

The bending stresses are computed as

$$\sigma_{M1} = \frac{M_1 x_2}{I_1}$$
$$\sigma_{M2} = -\frac{M_2 x_1}{I_2}$$

Where M_1 is the moment around the first principal axis and M_2 is the moment around the second principal axis.

BENDING SHEAR STRESSES

The components of the shear stresses caused by a shear force T_1 along the x_1-axis are

$$\tau_{T1x} = (\tau_{1z}^{(1)} \cos\alpha - \tau_{2z}^{(1)} \sin\alpha) \cdot T_1$$
$$\tau_{T1y} = (\tau_{2z}^{(1)} \cos\alpha + \tau_{1z}^{(1)} \sin\alpha) \cdot T_1$$

The components of the shear stresses caused by a shear force T_2 along the x_2-axis are

$$\tau_{T2x} = (\tau_{1z}^{(2)} \cos\alpha - \tau_{2z}^{(2)} \sin\alpha) \cdot T_2$$
$$\tau_{T2y} = (\tau_{2z}^{(2)} \cos\alpha + \tau_{1z}^{(2)} \sin\alpha) \cdot T_2$$

In both cases, the resultants are also computed as

$$\tau_{T1} = \sqrt{\tau_{T1x}^2 + \tau_{T1y}^2} \text{ and } \tau_{T2} = \sqrt{\tau_{T2x}^2 + \tau_{T2y}^2}$$

TORSIONAL SHEAR STRESSES

The components of the shear stress caused by Saint-Venant torsion are

$$\tau_{\mathrm{Mtx}} = \frac{M_{\mathrm{t}}\phi_{,\mathrm{y}}}{J}$$

$$\tau_{\mathrm{Mty}} = -\frac{M_{\mathrm{t}}\phi_{,\mathrm{x}}}{J}$$

where M_{t} is the twisting moment. The resultant is computed as

$$\tau_{\mathrm{Mt}} = \sqrt{\tau_{\mathrm{Mtx}}^2 + \tau_{\mathrm{Mty}}^2}$$

EQUIVALENT STRESS

The von Mises equivalent stress is computed from the stress components defined above using the expression

$$\sigma_{\mathrm{vM}} = \sqrt{(\sigma_N + \sigma_{\mathrm{M1}} + \sigma_{\mathrm{M2}})^2 + 3(\tau_{\mathrm{T1x}} + \tau_{\mathrm{T2x}} + \tau_{\mathrm{Mtx}})^2 + 3(\tau_{\mathrm{T1y}} + \tau_{\mathrm{T2y}} + \tau_{\mathrm{Mty}})^2}$$

Theory for the Truss Interface

General Theory for the Truss Interface

SHAPE FUNCTIONS

Trusses are modeled using Lagrange shape function. The Lagrange shape function makes it possible to specify both normal strains and Green–Lagrange strains to handle small strains as well as large deformations.

STRAIN–DISPLACEMENT RELATION

The axial strain ε_n is calculated by expressing the global strains in tangential derivatives and projecting the global strains on the edge.

$$\varepsilon_n = \mathbf{t}^t \varepsilon_{gT} \mathbf{t}$$

where $\mathbf{t}$ is the edge tangent vector and ε_{gT} is defined as

$$\varepsilon_{gT} = \begin{bmatrix} \varepsilon_{xT} & \varepsilon_{xyT} & \varepsilon_{xzT} \\ \varepsilon_{xyT} & \varepsilon_{yT} & \varepsilon_{yzT} \\ \varepsilon_{xzT} & \varepsilon_{yzT} & \varepsilon_{zT} \end{bmatrix}$$

The strains can be expressed as either engineering strains for small displacements or Green–Lagrange strains for large displacements. The Green–Lagrange strain tensor used for large displacements is defined as

$$\varepsilon_{ijT} = \frac{1}{2}\left(\left.\frac{\partial u_i}{\partial x_j}\right|_T + \left.\frac{\partial u_j}{\partial x_i}\right|_T + \left.\frac{\partial u_k}{\partial x_i}\right|_T \cdot \left.\frac{\partial u_k}{\partial x_j}\right|_T \right)$$

The engineering strain tensor used for small displacements is defined as

$$\varepsilon_{ijT} = \frac{1}{2}\left(\left.\frac{\partial u_i}{\partial x_j}\right|_T + \left.\frac{\partial u_j}{\partial x_i}\right|_T \right) \tag{3-269}$$

The axial strain written out becomes

$$\varepsilon_n = t_x(\varepsilon_{xT}t_x + \varepsilon_{xyT}t_y + \varepsilon_{xzT}t_z) + t_y(\varepsilon_{xyT}t_x + \varepsilon_{yT}t_y + \varepsilon_{yzT}t_z) + t_z(\varepsilon_{xzT}t_x + \varepsilon_{yzT}t_y + \varepsilon_{zT}t_z)$$

In the Truss interface, the coordinates are usually denoted with lowercase letters (x, y, z). If a Solid Mechanics or Membrane interface is present in the same model, then it becomes necessary to make a difference between the material frame and the spatial frame (Material and Spatial Coordinates). In this case the coordinates in the Truss interface change to (X, Y, Z).

STRESS–STRAIN RELATION

The constitutive relation for a truss is uniaxial. The axial stress, σ_n, is computed as

$$\sigma_n = \sigma_{ex} + E\varepsilon_{n,el} = \sigma_{ex} + E(\varepsilon_n - \varepsilon_{inel})$$

$$\varepsilon_{inel} = \varepsilon_0 + \varepsilon_{th} + \varepsilon_{hs} + \varepsilon_{pl}$$

$$\sigma_{ex} = \sigma_0 + \sigma_{ext}$$

where

- E is the modulus of elasticity
- ε_n is the total axial strain
- $\varepsilon_{n,el}$ is the elastic axial strain
- ε_{inel} is the sum of all inelastic strain contributions:
  - Initial strain, ε_0
  - Thermal strain, ε_{th}
  - Hygroscopic strain, ε_{hs}
  - Plastic strain, ε_{pl}
- σ_{ex} is the sum of all extra stress contributions:
  - Initial stress σ_0
  - External stress σ_{ext}

In a geometrically nonlinear analysis, this equation should be interpreted as a relation between Second Piola–Kirchhoff stresses and Green–Lagrange strains.

For output, the First Piola–Kirchhoff stress P_n is computed from the Second Piola–Kirchhoff stress using

$$P_n = S_n \cdot s'$$

where *s'* is the ratio between current and initial length. The axial force in the element is then computed as

$$N = P_n \cdot A_0$$

where A_0 is the undeformed cross-section area. The engineering (Cauchy) stress is defined by

$$\sigma_n = P_n \frac{A_0}{A}$$

where A is the deformed area of the element. For a geometrically linear analysis, the change in area is ignored, so that $A = A_0$.

For a geometrically nonlinear analysis, the area change is computed based on an assumption about a linear elastic material with Poisson's ratio ν. The area change is

$$\frac{A_0}{A} = (1 - 2\nu\varepsilon_n)^{-1}$$

This is the only occasion where the Truss interface makes use of the value of ν.

In a geometrically linear analysis all the stress representations have the same value.

IMPLEMENTATION

Using the principle of virtual work results in the following weak formulation

$$\delta W = d\int_V (-\varepsilon_n \sigma_n + \mathbf{u}^t \mathbf{F}_V) dV + \sum_i \mathbf{u}^t \mathbf{F}_{Pi}$$

where the summation stands for summation over all points in the geometry. Replacing the integration over the cross section with the cross-sectional area (A) and the volume forces with line forces, the equation becomes

$$\delta W = \int_L (-\varepsilon_{n\text{test}} \sigma_n A + \mathbf{u}^t_{\text{test}} \mathbf{F}_L) dL + \sum_i \mathbf{u}^t_{\text{test}} \mathbf{F}_{Pi}$$

In the case of geometric nonlinearity, the stress and strain should be interpreted as Second Piola–Kirchhoff stress and Green–Lagrange strains

THEORY FOR STRAIGHT EDGE CONSTRAINT

The optional constraint to enforce the nodes to lie on the straight line between the endpoints of the edge are formulated as follows:

Starting with the large displacement case, let $\mathbf{x}_{d1}$ and $\mathbf{x}_{d2}$ be the deformed position of the two endpoints of the edge

$$\mathbf{x}_{di} = \mathbf{u}_i + \mathbf{x}_i \tag{3-270}$$

where $\mathbf{u}_i$ is the displacement, and $\mathbf{x}_i$ is the coordinate (undeformed position) at endpoint i. The equation for the straight line through the endpoints is

$$\mathbf{x} + \mathbf{u} = \mathbf{x}_{d1} + t\mathbf{a} \tag{3-271}$$

where t is a parameter along the line, and $\mathbf{a}$ is the direction vector for the line. $\mathbf{a}$ is calculated from the deformed position of the endpoints as

$$\mathbf{a} = \mathbf{x}_{d2} - \mathbf{x}_{d1}$$

The constraints for the edge are derived by substituting the parameter t from one of the scalar equations in Equation 3-271 into the remaining ones. In 2D the constraint equations become

$$(x + u - x_{d1})a_y - (y + v - y_{d1})a_x = 0$$

In 3D the two constraints equations become

$$(x + u - x_{d1})a_z - (z + w - z_{d1})a_x = 0$$
$$(y + v - y_{d1})a_z - (z + w - z_{d1})a_y = 0$$

To avoid problems when the edge is directed in one of the coordinate axes directions, a third constraint is added. This constraint is a linear combination of the two earlier constraints:

$$(y + v - y_{d1})a_x - (x + u - x_{d1})a_y = 0$$

This constraint is nonlinear, since $\mathbf{a}$ depends on the displacement.

A linear constraint is needed in the case of a geometrically linear problem to become independent of the solver. The linear relation for the displacement is

$$\mathbf{u} = \frac{\mathbf{u}_1(x_{n2}-x_n)+\mathbf{u}_2(x_n-x_{n1})}{(x_{n2}-x_{n1})} + u_{ax}(\mathbf{x}_2-\mathbf{x}_1) \tag{3-272}$$

where u_{ax} is the axial displacement along the edge, and x_n is a linear parameter along the edge

$$x_n = \frac{x(x_2-x_1)+y(y_2-y_1)+z(z_2-z_1)}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}}$$

Eliminating u_{ax} from Equation 3-272 results in the following linear constraint in 2D

$$\left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](y_2-y_1) - \left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](x_2-x_1) = 0$$

and the following three linear constraints in 3D:

$$\begin{aligned}
&\left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](z_2-z_1) - \left[\frac{w_1(x_{n2}-x_n)+w_2(p-x_{n1})}{(x_{n2}-x_{n1})}-w\right](x_2-x_1) = 0\\
&\left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](z_2-z_1) - \left[\frac{w_1(x_{n2}-x_n)+w_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-w\right](y_2-y_1) = 0\\
&\left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](x_2-x_1) - \left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](y_2-y_1) = 0
\end{aligned} \tag{3-273}$$

THEORY FOR SPRING-DAMPER MATERIAL

Introduction

You can use a Spring-Damper Material to connect two points by an elastic spring, a viscous damper, or both. Such springs can be used in any structural mechanics physics interface, by adding a Truss interface. You can then set the degree of freedom names in the two interfaces to the same name, in order to share the same displacement fields.

The two points can move relative to each other in an arbitrary way as long as they do not coincide. The spring and damping forces act along the line between the current positions of the two connection points.

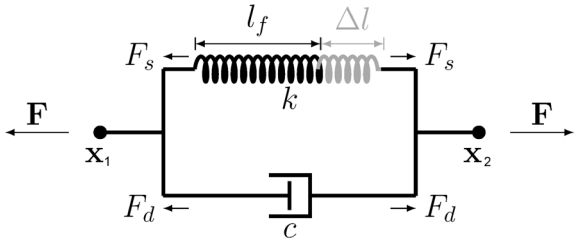


Figure 3-72: Conceptual sketch of the Spring-Damper Material.

Connection Points and Spring Length

The current position of the two endpoints, $\mathbf{x}_1$ and $\mathbf{x}_2$, can be written as

$$\mathbf{x}_1 = \mathbf{X}_1 + \mathbf{u}_1$$
$$\mathbf{x}_2 = \mathbf{X}_2 + \mathbf{u}_2$$

where $\mathbf{X}_1$ and $\mathbf{X}_1$ are the original positions of the two points, and $\mathbf{u}_1$ and $\mathbf{u}_2$ are their respective displacements. The initial spring length, l_0, is

$$l_0 = \|\mathbf{X}_2 - \mathbf{X}_1\| = \sqrt{(\mathbf{X}_2 - \mathbf{X}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1)}$$

The current spring length, l, is

$$l = \|\mathbf{x}_2 - \mathbf{x}_1\|$$

In the case of a geometrically linear analysis, the current spring length is linearized to

$$l = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1)}{l_0}$$

In addition to the initial geometrical distance between the two points you can specify an initial spring extension Δl_0, so that the free length of the spring is

$$l_\mathrm{f} = l_0 - \Delta l_0$$

You can also specify the free length of the spring explicitly.

The spring extension Δl is computed as the difference between the current spring length and the free length,

$$\Delta l = l - l_{\mathrm{f}}$$

Deactivation

You can specify that the Spring-Damper Material should be deactivated under certain conditions. It can for example be active only in tension, or break at a certain elongation. In terms of implementation, this means that many expressions are multiplied by an *activation indicator*, i_{ac}. The activation indicator has the value 1 when the component is active, and 0 when deactivated.

Spring and Damping Forces

The spring force is proportional to the spring constant k:

$$F_{\mathrm{s}} = k\Delta l$$

If k depends on the extension, so that the spring is nonlinear, it should be interpreted as a secant stiffness, that is

$$F_{\mathrm{s}} = k(\Delta l) \cdot \Delta l$$

You can also specify the spring force as function of extension explicitly as

$$F_{\mathrm{s}} = F_{\mathrm{s}}(\Delta l)$$

To create the expression for the function, use the built-in variable for the spring extension. It has the form `<physicsTag>.<SpringNodeTag>.dl`, for example `truss.spd1.dl`.

In a dynamic analysis, the viscous damping force is computed as

$$F_{\mathrm{d}} = c\frac{d}{dt}(\Delta l)$$

where c is the viscous damping coefficient. In frequency domain, it is also possible to specify a loss factor η, and the total damping force will then be

$$F_{\mathrm{d}} = i\omega c\Delta l + i\eta F_{\mathrm{s}}$$

The magnitude of the total force is

$$F = F_{\mathrm{s}} + F_{\mathrm{d}}$$

The total forces in the global coordinate system, acting on the connection points are

$$\mathbf{F}_2 = -\mathbf{F}_1 = -F\frac{(\mathbf{x}_2 - \mathbf{x}_1)}{l}$$

In a geometrically linear case, the orientation of the force is kept fixed, so that

$$\mathbf{F}_2 = -\mathbf{F}_1 = -F\frac{(\mathbf{X}_2 - \mathbf{X}_1)}{l}$$

The contribution to the virtual work is

$$\mathbf{F}_2 \cdot (\delta\mathbf{u}_2 - \delta\mathbf{u}_1)$$

Spring and Damping Energies

In stationary and time-dependent analysis, the elastic energy in the spring is computed as

$$W_s = i_{ac}\int_0^{\Delta l} F_s d(\Delta l)$$

In a time-dependent analysis, the energy dissipated in the damper, W_d, is computed using an extra degree of freedom. The following equation is added:

$$\frac{dW_d}{dt} = i_{ac}F_d\frac{d(\Delta l)}{dt}$$

In a frequency domain analysis, the elastic energy in the spring and the energy dissipated in the damper are computed as

$$W_s = \frac{1}{4}i_{ac}\mathrm{Re}(\mathrm{lindev}(F_s)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

$$W_d = \frac{1}{4}i_{ac}\mathrm{Re}(\mathrm{lindev}(F_d)\cdot\mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

These energy quantities represent the cycle average, and only the perturbation terms are included.

LOCAL BUCKLING

In a truss model, each individual mesh element is assumed to be straight. A linear buckling analysis gives the buckling load for entire truss structure, but it cannot detect local buckling of individual members.

The buckling load for each member can however be directly computed from material and geometrical properties, and variables for this are available.

The critical load of an individual bar can be expressed as

$$F_c = \frac{\pi^2 E I_{min}}{(KL)^2} \tag{3-274}$$

Here, E is Young's modulus, I_{min} is the area moment of inertia in the weakest orientation of the cross section, and L is the length of the bar. K is called the *effective length factor*, and can be used to compensate for nonideal end conditions. The value 1 corresponds to the Euler 2 buckling case, in which both ends are pinned.

The failure index is computed as the ratio between the compressive axial force N and the critical load:

$$f_i = \frac{-N}{F_c} \tag{3-275}$$

A failure index exceeding 1 means that the buckling load is exceeded. A set of variables providing different measures of the utilization is listed in Table 3-19.

TABLE 3-19: VARIABLES FOR LOCAL BUCKLING.

	DEFINITION	DESCRIPTION
truss.lbFc	Equation 3-274	Local buckling critical force
truss.lbf_i	Equation 3-275	Local buckling failure index
truss.lbs_f	1/truss.lbf_i	Local buckling safety factor
truss.lbm_s	truss.lbs_f - 1	Local buckling margin of safety
truss.lbd_i	if(truss.lbf_i>1, 1, 0)	Local buckling damage index

Theory for the Wire Interface

General Theory for the Wire Interface

SHAPE FUNCTIONS

The displacements in the Wire interface are represented by Lagrange shape functions.

STRAIN–DISPLACEMENT RELATION

The axial strain ε_n is calculated by expressing the global strains in tangential derivatives and projecting the global strains on the edge.

$$\varepsilon_n = \mathbf{t}^t \varepsilon_{gT} \mathbf{t}$$

where $\mathbf{t}$ is the edge tangent vector and ε_{gT} is defined as

$$\varepsilon_{gT} = \begin{bmatrix} \varepsilon_{xT} & \varepsilon_{xyT} & \varepsilon_{xzT} \\ \varepsilon_{xyT} & \varepsilon_{yT} & \varepsilon_{yzT} \\ \varepsilon_{xzT} & \varepsilon_{yzT} & \varepsilon_{zT} \end{bmatrix}$$

The strains is expressed as Green–Lagrange strains, allowing large displacements and rotations.

$$\varepsilon_{ijT} = \frac{1}{2}\left(\left.\frac{\partial u_i}{\partial x_j}\right|_T + \left.\frac{\partial u_j}{\partial x_i}\right|_T + \left.\frac{\partial u_k}{\partial x_i}\right|_T \cdot \left.\frac{\partial u_k}{\partial x_j}\right|_T\right)$$

The axial strain written out becomes

$$\begin{aligned} \varepsilon_n = {} & t_x(\varepsilon_{xT} t_x + \varepsilon_{xyT} t_y + \varepsilon_{xzT} t_z) + \\ & t_y(\varepsilon_{xyT} t_x + \varepsilon_{yT} t_y + \varepsilon_{yzT} t_z) + \\ & t_z(\varepsilon_{xzT} t_x + \varepsilon_{yzT} t_y + \varepsilon_{zT} t_z) \end{aligned}$$

STRESS–STRAIN RELATION

The constitutive relation for a wire is uniaxial. Since wires in most cases do not have a homogeneous cross section, it is not meaningful to compute a stress. Rather, a force versus strain relation is used. The axial force in tension, N, is computed as

$$N = N_{\mathrm{ex}} + k_{\mathrm{A}} \varepsilon_{\mathrm{n,\,el}} = N_{\mathrm{ex}} + k_{\mathrm{A}}(\varepsilon_{\mathrm{n}} - \varepsilon_{\mathrm{inel}})$$

$$\varepsilon_{inel} = \varepsilon_0 + \varepsilon_{th} + \varepsilon_{hs}$$

$$\sigma_{ex} = \sigma_0 + \sigma_{ext}$$

where

- k_A is the axial stiffness of the wire
- ε_n is the total axial strain
- $\varepsilon_{n,el}$ is the elastic axial strain
- ε_{inel} is the sum of all inelastic strain contributions, such as
  - Initial strain, ε_0
  - Thermal strain, ε_{th}
  - Hygroscopic strain, ε_{hs}
- N_{ex} is the sum of all extra axial force contributions such as
  - Initial axial force N_i
  - External axial force N_{ext}

In a geometrically nonlinear analysis, the strains are interpreted as Green–Lagrange strains. It should be noted that while Green–Lagrange strains are formally not additive, it assumed that even if displacements and rotations are large, the axial strains are small.

The ideal wire will not be able to sustain compressive forces. In practice, it may wrinkle in an unpredictable manner when tension is lost. In order to maintain numerical stability, a small stiffness is instead used when in compression. The following expression is used for the force in this case:

$$N = \beta k_A \left[\varepsilon_{n,el} - \varepsilon_{n,c} + \gamma\left(1 - \frac{1}{\beta}\right) \right] \qquad \varepsilon_{n,el} < \varepsilon_{n,c}$$

$$N = \gamma k_A \left(e^{\frac{\varepsilon_{n,el}}{\gamma}} - 1 \right) \qquad \varepsilon_{n,c} < \varepsilon_{n,el} < 0$$

where β is a stiffness reduction factor, and

$$\gamma = \frac{\varepsilon_{n,c}}{\ln \beta}$$

The reduced compressive stiffness is constant, βk_A, below a certain compressive strain $\varepsilon_{n,c}$. Between the two domains of constant stiffness, there is a transition region in

which the stiffness drops exponentially, so that the strain–force relation always has a continuous derivative.

IMPLEMENTATION

Using the principle of virtual work results in the following weak formulation

$$\delta W = d\int_V (-\varepsilon_n \sigma_n + \mathbf{u}^t \mathbf{F}_V) dV + \sum_i \mathbf{u}^t \mathbf{F}_{Pi}$$

where the summation stands for summation over all points in the geometry. Here, stresses are not directly accessible. The volume integration must be replaced by a line integration.

$$\delta W = \int_L (-\varepsilon_{n\text{test}} N + \mathbf{u}^t_{\text{test}} \mathbf{F}_L) dL + \sum_i \mathbf{u}^t_{\text{test}} \mathbf{F}_{Pi}$$

Theory for the Pipe Mechanics Interface

The Pipe Mechanics Interface theory is described in this section:

- About Pipes
- Effects of Internal Pressure
- Effects of Thermal Expansion
- Stress Evaluation
- Connection Between Pipes and Structures
- Cross Sections

About Pipes

The Pipe Mechanics interface is intended for modeling slender pipes with arbitrary cross sections. It is based on beam theory, and the similarities with the Beam interface are large. The most important features that make the Pipe Mechanics interface different from the Beam interface are:

- In a pipe, the hoop stress caused by the internal pressure gives a significant contribution to the stress state.
- The hoop stress will cause a contraction of the pipe in the axial direction due to Poisson's ratio effects.
- The mass of the enclosed fluid must be taken into account when computing gravity loads and dynamic properties. This effect is active only in the transverse direction of the pipe, but not in the axial direction.
- When working with thermal expansion, the temperature is assumed to have one value at the inside of the pipe and another value at the outside. There is thus no global bending temperature gradient causing a net moment over the cross section. Rather, there is local thermal stress gradient through the pipe wall.

Most of the theory of the Pipe Mechanics interface is shared with the Beam Interface.

In the theory for the Beam interface:

- Shape Functions
- Geometric Nonlinearity
- Strain–Displacement/Rotation Relation
- Stress–Strain Relation
- Initial Stress and Strain
- Geometric Variables

Only the parts of the theory which is specific to pipes will be covered here. Since the by far most common cross section is a circular pipe, this geometry is on several occasions given a special treatment.

Effects of Internal Pressure

HOOP STRESS

When an internal pressure is present in a pipe, it must be balanced by a net force in the circumferential direction. For a circular pipe, this stress component is often called *hoop stress*. For a thin-walled circular pipe, it is given by the well-known expression

$$\sigma_h = \frac{pR}{t}$$

where p is the pressure, R is the radius of the pipe, and t is the wall thickness.

The pressure p here is the *gauge pressure*, assumed to be positive. If the pressure in the pipe is given as an *absolute pressure*, then the external pressure (usually 1 atm) must also be supplied. It is assumed that the external pressure is small relative to the internal pressure, so that the stress distribution through the pipe wall is given only by the gauge pressure $p = p_{\text{internal}} - p_{\text{external}}$.

In particular, the Pipe Flow interface operates with absolute pressures, so when running a combined analysis, it is important to include also the external pressure.

For stress evaluation in a general cross section, the average (or membrane) hoop stress can be written as

$$\sigma_{h,\,m} = \gamma_m p$$

where γ_m is a dimensionless factor multiplying the pressure.

The average hoop stress is available in the variable `pipem.shm`.

For a general thick-walled circular pipe, the average hoop stress is given by

$$\gamma_m = \frac{R}{t} = \frac{d_i}{d_o - d_i}$$

where the inner and outer diameters have been introduced.

For a thick circular pipe, the hoop stress may however have a significant variation through the thickness (Figure 3-73).

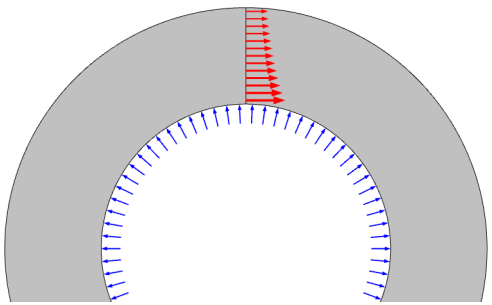


Figure 3-73: Hoop stress distribution in a thick-walled pipe.

The peak circumferential stress is

$$\sigma_{h,\,max} = p\frac{1 + \left(\frac{d_i}{d_o}\right)^2}{1 - \left(\frac{d_i}{d_o}\right)^2}$$

The difference between the peak circumferential stress and the mean circumferential stress is

$$\sigma_{h,\,b} = \frac{p}{1 + \frac{d_i}{d_o}} = \gamma_b p$$

Here, γ_b is a cross section dependent factor, which is used to compute the peak stress due to internal pressure.

The bending part of the hoop stress is available in the variable `pipem.shb`.

For noncircular pipe sections, the nonuniform stress distribution is more prominent. For such sections, the bending stress in the pipe wall can be much larger than the mean stress which balances the internal pressure. This is indicated in Figure 3-74.

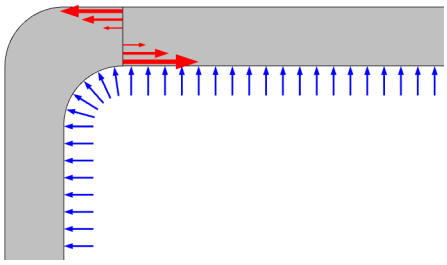


Figure 3-74: Bending stress caused by internal pressure in rectangular pipe section.

For a rectangular pipe section, the bending stress can be estimated using beam theory. If the cross section is considered as a rectangular frame with constant thickness t, then the moment (per unit length) at the corner will be

$$M = \frac{p}{12} \cdot \frac{L_y^3 + L_z^3}{L_y + L_z}$$

Here, the length in the horizontal direction is L_y and the length in the vertical direction is L_z.

The nominal bending stress at the corner is thus

$$\sigma_{\mathrm{h,\,b}} = \frac{6M}{t^2} = \frac{p}{2t^2} \cdot \frac{L_y^3 + L_z^3}{L_y + L_z}$$

The bending part of the hoop stress is available in the variable `pipem.shb`.

For normal values of wall thickness to section width ratios and corner fillet radii, this will give a good approximation.

The average stress, which is given by the force balance, is

$$\sigma_{\mathrm{h,\,m}} = p\frac{\max(L_y, L_z)}{t}$$

The average hoop stress is available in the variable `pipem.shm`.

Thus,

$$\gamma_{\mathrm{m}} = \frac{\max(L_y, L_z)}{t}$$

$$\gamma_{\mathrm{b}} = \frac{1}{2t^2} \cdot \frac{L_y^3 + L_z^3}{L_y + L_z}$$

For rectangular pipe sections, the bending stress is thus always larger than the membrane stress.

POISSON'S RATIO EFFECT

Due to the stresses and corresponding strains in the circumferential direction, there will be a coupling to axial deformation through Poisson's ratio. The bending stresses within the pipe wall are self-equilibrating and does not have a net effect. For a cylindrical pipe, it can be shown that the radial and hoop stresses together cause a homogeneous axial strain throughout the wall. Thus, no local axial stresses are introduced. For a general cross section, the local axial stress caused by is assumed to be

$$\sigma_{n,\,b} = \nu\sigma_{h,\,b}$$

The axial stress contribution from the bending part of the hoop stress is available in the variable `pipem.shbn`.

The average hoop stress will, however, always cause an axial strain. Since the diameter of the pipe increases under an internal pressure, the pipe must, if the ends are free, contract. Equivalently, if the ends of the pipe are kept fixed, a tensile axial force is introduced. It can be shown that for any pipe cross section, this force is

$$N = 2\nu p A_f$$

where A_f is the cross-section area of the fluid. Note that the force is independent of the geometry of the wall itself.

The corresponding axial strain in the case of free axial deformation is

$$\varepsilon_{ap} = \frac{-2\nu p A_f}{EA}$$

This term is added as an initial strain to the constitutive relation.

Fluid Flow Forces

In addition to the pressure expanding the pipe, there can be loads caused by momentum transfer. These loads are of several types:

- Drag forces due to friction between the fluid and the pipe wall
- Centrifugal forces due to redirection of the flow in pipe bends
- Pressure forces in pipe bends
- Point forces caused by for example pressure drops in valves and changes in pipe cross section

DRAG FORCE

The drag force, $\mathbf{F}_d$, always acts along the tangent of the pipe in the direction of the flow, $\mathbf{t}$. Its contribution to the edge load per unit length is

$$\mathbf{F}_d = \frac{1}{2} f_D \rho_f A_f u_f |u_f| \mathbf{t}$$

where the subscript 'f' stands for fluid properties. The Darcy friction factor f_D is a dimensionless coefficient that is specified in the **Fluid Properties** node in the Pipe Flow interface. For most friction models, it is effectively proportional to $1/\mathrm{Re}$, where $\mathrm{Re} = \rho_f |u_f| d_h / \mu_f$ is the local Reynolds number in the flow. Thus, the resulting magnitude of the drag force becomes proportional to the velocity.

In the Pipe Flow interface, $\mathbf{F}_d$ is computed and is made available for the Pipe Mechanics interface.

CENTRIFUGAL FORCE

The centrifugal force contribution to the edge load per unit length is

$$\mathbf{F}_c = -\rho_f A_f u_f^2 \mathbf{k}_n$$

where the curvature vector is computed as a derivative of the edge tangent with respect to the length parameter

$$\mathbf{k}_n = \frac{\partial \mathbf{t}}{\partial s}$$

This vector is perpendicular to the tangent, and it points toward the local center of curvature. The centrifugal force acts in the direction opposite to the vector.

In the Pipe Flow interface, $\mathbf{F}_c$ is computed and is made available for the Pipe Mechanics interface.

PRESSURE FORCE

In curved pipes, the pressure force contribution to the edge load per unit length is

$$\mathbf{F}_p = -(p_f - p_o) A_f \mathbf{k}_n$$

where p_o is the exterior pressure on the outside of the pipe.

In the Pipe Flow interface, $\mathbf{F}_p$ is computed and is made available for the Pipe Mechanics interface.

JUNCTION POINT FORCES

Point loads are related to the momentum change due to the flow alteration at junction points. For each junction, the force can be computed as a jump in the quantity $(p_f - p_o + \rho_f u_f^2) A_f \mathbf{t}$. Thus, the resulting point load can be written as

$$\mathbf{F}_{\mathrm{jp}} = \sum_n s_n[(p_\mathrm{f} - p_\mathrm{o} + \rho_\mathrm{f} u_\mathrm{f}^2)A_\mathrm{f}\mathbf{t}]_n$$

where the summation is performed over all pipe ends at the junction point, and s_n stands for either plus or minus sign depending on the flow direction.

You need to use the functionality available on the Fluid–Pipe Interaction coupling node to apply the junction forces.

Effects of Thermal Expansion

Temperature changes in the pipe have several effects. A homogeneous change in temperature will cause the pipe to extend or shrink in the axial direction. When there is a temperature gradient through the pipe wall, the mean temperature will control this deformation in an average sense. There will, however, also be local strain and stress states caused by the temperature variation. These stresses can be significant.

For analysis of cases with thermal expansion, it is assumed that the inside of the pipe has a temperature T_i, which is constant along the perimeter. As a first approximation, this would be the temperature of the fluid. In reality, there is a temperature jump given by the heat transfer coefficient. Similarly, it is assumed that the outside of the pipe has a temperature T_o, which is constant along the perimeter. Note that if there is insulation around the pipe, T_o represents the temperature of the pipe wall inside the insulation.

The variation of the temperature through the wall is treated slightly differently for circular cross section and in the general case. In the circular pipe, the temperature has a distribution which can be determined from analytically.

GENERAL CROSS SECTION

For a general cross section, it is assumed that the temperature varies linearly through the pipe wall, and that the average temperature is

$$T_\mathrm{m} = \frac{T_\mathrm{i} + T_\mathrm{o}}{2}$$

If the coefficient of thermal expansion is temperature dependent, it is assumed that its variation through the pipe wall is linear.

Define

$$\alpha_i = \alpha(T_i)$$
$$\alpha_o = \alpha(T_o)$$
$$\Delta\alpha = \alpha_o - \alpha_i$$
$$\Delta T = T_o - T_i$$

The average axial thermal strain is then

$$\varepsilon_{th} = \alpha_i(T_i - T_{ref}) + \frac{1}{2}(\Delta\alpha(T_i - T_{ref}) + \alpha_i\Delta T) + \frac{1}{3}\Delta\alpha\Delta T$$

CIRCULAR PIPE

For a circular pipe, the heat transfer problem can be solved analytically. The radial temperature distribution is given by

$$T(r) = T_i + \Delta T \frac{\ln\frac{r}{r_i}}{\ln\frac{r_o}{r_i}}$$

As long as the wall thickness is small, the temperature distribution through the thickness is close to linear.

If the coefficient of thermal expansion is temperature dependent, the exact axial strain has to be computed using an integral through the thickness of the material. In order to avoid performing such integration in runtime, a linear variation of the coefficient of thermal expansion with the radial coordinate is assumed,

$$\alpha(r) = \alpha_i + \frac{\Delta\alpha(r - r_i)}{r_o - r_i}$$

The averaged axial thermal strain is then

$$\varepsilon_{th} = \frac{2}{r_o^2 - r_i^2}\int_{r_i}^{r_o} \alpha(r)(T(r) - T_{ref})r\,dr$$

Evaluation of the integral gives

$$\varepsilon_{th} = \alpha_i(T_i - T_{ref}) + \frac{1}{3}\Delta\alpha(T_i - T_{ref})\left(1 + \frac{r_o}{r_o + r_i}\right) +$$

$$\alpha_i \Delta T\left(\frac{r_o^2}{r_o^2 - r_i^2} - \frac{1}{2\log\left(\frac{r_o}{r_i}\right)}\right) +$$

$$\Delta\alpha\Delta T\left(\frac{r_o^2(2r_o - 3r_i)}{3(r_o - r_i)^2(r_o + r_i)} - \frac{4r_o^2 - 5r_i^2 - 5r_o r_i}{18(r_o^2 - r_i^2)\log\left(\frac{r_o}{r_i}\right)}\right)$$

BENDING THERMAL STRESS

The temperature gradient through the pipe wall causes a local bending stress state, both in the axial direction and in the hoop direction. This contribution to the stress is taken into account only for the circular pipe section.

The peak value of the bending stress is calculated as

$$\sigma_{t,nb} = \sigma_{t,hb} = \frac{(\alpha_i + \alpha_o)\Delta T}{2(1 - \upsilon)}\left[\frac{r_o^2}{r_o^2 - r_i^2} - \frac{1}{2\log\left(\frac{r_o}{r_i}\right)}\right]$$

Stress Evaluation

Since the basic result quantities for beams are the integrated stresses in terms of section forces and moment, special considerations are needed for the evaluation of actual stresses. When combining various stress contributions, a conservative approach is taken in every step.

AXIAL STRESS

The axial stress in the pipe has the following contributions:

- Axial force
- Bending moment
- Local bending stress caused by internal pressure
- Local bending stress caused by temperature gradient through the wall.

Normal Force

The normal stress from axial force is constant over the section, and computed as

$$\sigma_n = \frac{N}{A}$$

Bending Moments

The normal stress from bending is computed in four user-selected points (y_{lk}, z_{lk}) in the cross section as

$$\sigma_{bk} = \frac{M_{ly} z_{lk}}{I_{yy}} - \frac{M_{lz} y_{lk}}{I_{zz}}$$

In 2D, only two points, specified by their local y-coordinates are used.

Internal Pressure

The effects of internal pressure are described in detail in Effects of Internal Pressure above. The axial bending stress component $\sigma_{n,b}$ is always positive due to the assumption that the gauge pressure is positive. In reality, it represents a bending stress, so the true sign differs between the inside and the outside of the pipe.

Thermal Expansion

The effects of thermal expansion are described in detail in Effects of Thermal Expansion above. The axial bending stress component $\sigma_{t,b}$ is always positive due its definition. In reality, it represents a bending stress, so the true sign differs between the inside and the outside of the pipe.

Total Axial Stress

The total axial stress in evaluation point k is defined as

$$\sigma_k = \sigma_{bk} + \sigma_n + (\sigma_{n,b} + \sigma_{t,nb})\text{sign}(\sigma_{bk} + \sigma_n)$$

If the true distributions of $\sigma_{n,b}$ and $\sigma_{t,b}$ have opposite signs, this will be conservative.

The peak normal stress in the section is defined as

$$\sigma_{n,\,max} = \max(|\sigma_k|)$$

A special method is used for the Pipe cross section. Since there are no extreme positions around a circle, a maximum bending stress is computed as

$$\sigma_{b,\,max} = \frac{d_o \sqrt{M_{ly}^2 + M_{lz}^2}}{2 I_{zz}}$$

where d_o is the outer diameter. This value replaces the stress from the stress evaluation points (σ_{bk}) in maximum stress expressions, thus ensuring that the correct peak stress is evaluated irrespective of where it appears along the circumference.

The following variables are used for the axial stress:

- Stress caused by axial force, σ_n: `pipem.sn`
- Stress caused by bending moments, σ_{bk}(evaluation points 1-4): `pipem.sb1`, ..., `pipem.sb4`
- Axial stress from gauge pressure, $\sigma_{n,b}$: `pipem.shb`
- Axial stress from temperature gradient, $\sigma_{t,b}$: `pipem.shb`
- Total axial stress in evaluation points, σ_k: `pipem.s1`, ..., `pipem.sb4`
- Maximum axial stress, σ_{max}: `pipem.smax`

HOOP STRESS

The hoop stress acts in a direction which is orthogonal to the axial stress. Hoop stresses can be caused by the internal pressure and by a temperature gradient through the thickness.

Internal Pressure

The hoop stress caused by the pressure is

$$\sigma_{h,p} = \sigma_{h,m} + \sigma_{h,b} = p(\gamma_m + \gamma_b)$$

where γ_m and γ_b are defined in Effects of Internal Pressure. The bending part is defined as positive. In reality, it represents a bending stress, so the true sign differs between the inside and the outside of the pipe. Under the assumption of a positive gauge pressure, the positive sum will however exist on either the inner or the outer boundary.

Thermal Expansion

The temperature gradient stress is caused by difference in thermal strain between the inside and the outside. The details are discussed in Effects of Thermal Expansion.

The total hoop stress is computed as

$$\sigma_{h,max} = \sigma_{h,p} + \sigma_{t,hb}$$

SHEAR STRESS

The shear stress from twist in general has a complex distribution over the cross section. The maximum shear stress due to torsion is defined as

$$\tau_{\mathrm{t,\,max}} = \frac{|M_{1x}|}{W_{\mathrm{t}}}$$

where W_{t} is the torsional section modulus. This result is available only in 3D.

The section shear forces are computed in two different ways depending on the beam formulation. For Euler–Bernoulli theory, the section forces proportional to the third derivative of displacement, or equivalently, the second derivative of the rotation.

$$T_{1z} = EI_{yy}\frac{\partial^2 \theta_{1y}}{\partial s^2}$$

$$T_{1y} = -EI_{zz}\frac{\partial^2 \theta_{1z}}{\partial s^2}$$

where T_{1z} is available only in 3D. In the case of Timoshenko theory shear force is computed directly from the shear strain.

The average shear stresses are computed from the shear forces as

$$\begin{aligned} \tau_{sz,\,\mathrm{ave}} &= \frac{T_{1z}}{A} \\ \tau_{sy,\,\mathrm{ave}} &= \frac{T_{1y}}{A} \end{aligned} \tag{3-276}$$

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\begin{aligned} \tau_{sz,\,\mathrm{max}} &= \mu_z \tau_{sz,\,\mathrm{ave}} \\ \tau_{sy,\,\mathrm{max}} &= \mu_y \tau_{sy,\,\mathrm{ave}} \end{aligned} \tag{3-277}$$

As the directions and positions of maximum shear stresses from shear and twist are not known in a general case, upper bounds to the shear stress components are defined as

$$\begin{aligned} \tau_{xz,\,\mathrm{max}} &= |\tau_{sz,\,\mathrm{max}}| + \tau_{\mathrm{t,\,max}} \\ \tau_{xy,\,\mathrm{max}} &= |\tau_{sy,\,\mathrm{max}}| + \tau_{\mathrm{t,\,max}} \end{aligned}$$

EQUIVALENT STRESS

The maximum von Mises equivalent stress for the cross section is then defined as

$$\sigma_{\text{mises}} = \sqrt{\sigma_{\text{n, max}}^2 + \sigma_{\text{h, max}}^2 + 3\tau_{xy,\,\text{max}}^2 + 3\tau_{xz,\,\text{max}}^2}$$

Since the maximum values for the different stress components in general occur at different positions in the cross section, the equivalent stress thus computed is a conservative approximation.

The Tresca equivalent stress is computed from the principal stresses. Determining the principal stresses is however a nontrivial operation. The first assumption is that the through-thickness direction is a principal orientation with zero principal stress. This is consistent with ignoring the radial stress in the von Mises stress expression.

For the remaining axial-hoop plane a Mohr's circle argument is used. The two nonzero principal stresses in that plane are computed for evaluation point k as

$$\sigma_{1,\,2}^k = \frac{1}{2}[\sigma_k + \sigma_{\text{h,max}} \pm \sqrt{(\sigma_k - \sigma_{\text{h,max}})^2 + 4(\tau_{xy,\,\text{max}}^2 + \tau_{xz,\,\text{max}}^2)}]$$

The principal stresses are ordered in descending order, and the Tresca stress is obtained as the highest value in any of the evaluation points.

$$\sigma_{\text{tresca}} = \max(\sigma_1^k - \sigma_3^k)$$

The following variables are used for the equivalent and principal stresses:

- von Mises equivalent stress, σ_{mises}: `pipem.mises`
- Tresca equivalent stress, σ_{tresca}: `pipem.tresca`
- First principal stress, σ_{1k} (evaluation points 1-4): `pipem.sp1_1`, …, `pipem.sp1_4`
- First principal stress, σ_{2k} (evaluation points 1-4): `pipem.sp2_1`, …, `pipem.sp2_4`
- First principal stress, σ_{3k} (evaluation points 1-4): `pipem.sp3_1`, …, `pipem.sp3_4`

Connection Between Pipes and Structures

This section describes the theory and assumption behind the Structure–Pipe Connection multiphysics coupling. The coupling is an extension of the transition type couplings in Solid–Beam Connection and Shell–Beam Connection to also account for radial deformation of the pipe caused by the fluid pressure and the temperature distribution over the cross section. For a more general background to the coupling of beam type elements to solids and shells, see Connection Between Shells and Beams and Connection Between Shells and Solids.

When connecting a pipe to a solid domain it is assumed that the pipe cross section is circular, and that no warping occurs. The connection thus, on the intersecting boundaries of the solid domain, adds the following constraint equation

$$\mathbf{u}_s = \mathbf{u}_p + \theta_p \times \mathbf{r} - \nu_s\left(\int \varepsilon_{x1,s} dy_1\right)\mathbf{e}_{y1} - \nu_s\left(\int \varepsilon_{x1,s} dz_1\right)\mathbf{e}_{z1} - \mathbf{u}_{radial} \qquad (3\text{-}278)$$

where $\mathbf{u}_s$ is the displacement of the solid, ν_s is Poisson's ratio of the solid, $\mathbf{u}_p$ is the displacement of the pipe, θ_p is the rotation of the pipe, $\mathbf{r}$ the distance from the center of the pipe, and $\mathbf{e}_{y1}$ and $\mathbf{e}_{z1}$ are the base vectors of the local y_1-axis and z_1-axis. The first four terms in Equation 3-278 are identical to the transition type coupling in the Solid–Beam Connection, while the last one is added to account for the fluid pressure and the temperature difference in the pipe. The radial displacement $\mathbf{u}_{radial}$ included in Equation 3-278 is computed from the radial deformation u_r of the pipe as

$$\mathbf{u}_{radial} = \mathbf{T}^T \cdot \left\{0, \frac{u_r d_y}{R}, \frac{u_r d_z}{R}\right\}$$

where $\mathbf{T}$ is a matrix the describes the transformation from the local coordinates of the pipe to global coordinates. Above, d_y and d_z are the distances, in local coordinates, and R is the radial distance from the center of the pipe. The radial displacement is given by

$$u_r = \frac{R}{E_s}[S_\theta - \nu_s(S_R + S_Z)] + R\alpha_p\left[T_{in} - T_{ref} + (T_{out} - T_{in})\frac{\log\left(\frac{R}{R_i}\right)}{\log\left(\frac{R_o}{R_i}\right)}\right] \qquad (3\text{-}279)$$

where E_s is Young's modulus of the solid, α_p is its coefficient of thermal expansion, R_i and R_o are its inner and outer radii, and T_{ref} is its volume reference temperature. The

stresses in the pipe due to the temperature difference between the inside and outside temperatures T_{in} and T_{out}, and the fluid pressure p on the inside surface of the pipe are given by the following analytical expressions

$$S_R = \frac{p\left(\frac{R_i}{R_o}\right)^2}{1-\left(\frac{R_i}{R_o}\right)^2}\left(1-\frac{R_o^2}{R^2}\right) + (T_{out}-T_{in})\alpha_p \frac{E_s}{2(1-\nu_s)}\left[\frac{1-\left(\frac{R_i}{R}\right)^2}{1-\left(\frac{R_i}{R_o}\right)^2} - \frac{\log\left(\frac{R}{R_i}\right)}{\log\left(\frac{R_o}{R_i}\right)}\right]$$

$$S_\theta = \frac{p\left(\frac{R_i}{R_o}\right)^2}{1-\left(\frac{R_i}{R_o}\right)^2}\left(1+\frac{R_o^2}{R^2}\right) + (T_{out}-T_{in})\alpha_p \frac{E_s}{2(1-\nu_s)}\left[\frac{1+\left(\frac{R_i}{R}\right)^2}{1-\left(\frac{R_i}{R_o}\right)^2} - \left(1+\frac{\log\left(\frac{R}{R_i}\right)}{\log\left(\frac{R_o}{R_i}\right)}\right)\right]$$

$$S_Z = (T_{out}-T_{in})\alpha_p \frac{E_s}{2(1-\nu_s)}\left[\frac{2}{1-\left(\frac{R_i}{R_o}\right)^2} - \left(1+\frac{2\log\left(\frac{R}{R_i}\right)}{\log\left(\frac{R_o}{R_i}\right)}\right)\right]$$

When the structure comes from a Shell interface, additional constraints are added for the rotational degrees of freedom as is done for the Shell–Beam connection, so that

$$\theta_p \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 = 0$$
$$\theta_p \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 = 0$$

where $\mathbf{a}$ is the shell normal displacement, and $\mathbf{t}_1$ and $\mathbf{t}_1$ are the shell tangents. Also, the expression for the radial deformation u_r in Equation 3-279 is simplified, and it is for a shell connection given by

$$u_r = \frac{R}{E_s}[S_\theta - \nu_s(S_R + S_Z)] + R\alpha(T_{in} - T_{ref})$$

Moreover, the expressions for stresses in the pipe simplifies to the thin pressure vessel approximation, so that

$$S_R = 0$$

$$S_\theta = s_p \frac{pR_i}{d} + s_T(T_{out} - T_{in})\alpha \frac{E_s}{2(1-\nu_s)}\left[\frac{2}{1-\left(\frac{R_i}{R_o}\right)^2} - \frac{1}{\log\left(\frac{R_o}{R_i}\right)}\right]$$

$$S_Z = s_T(T_{out} - T_{in})\alpha \frac{E_s}{2(1-\nu_s)}\left[\frac{2}{1-\left(\frac{R_i}{R_o}\right)^2} - \frac{1}{\log\left(\frac{R_o}{R_i}\right)}\right]$$

Here s_p and s_T are scaling factors, which are necessary to avoid abrupt changes in $\mathbf{u}_{\text{radial}}$.

Cross Sections

CIRCULAR SECTION

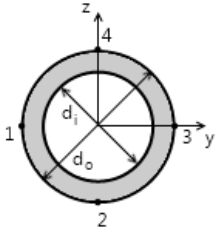


Figure 3-75: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-20: CIRCULAR SECTION CONSTANTS.

PROPERTY	FORMULA	REMARKS
A	$\frac{\pi(d_o^2 - d_i^2)}{4}$	
I_{zz}	$\frac{\pi(d_o^4 - d_i^4)}{64}$	

TABLE 3-20: CIRCULAR SECTION CONSTANTS.

PROPERTY	FORMULA	REMARKS
e_z	0	
μ_y	$\frac{(d_o^3 - d_i^3)A}{12(d_o - d_i)I_{zz}}$	
κ_y	$1 + \frac{d_i}{d_o}$	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	
κ_z	κ_y	
J	$\frac{\pi(d_o^4 - d_i^4)}{32}$	
W_t	$\frac{\pi(d_o^4 - d_i^4)}{16 d_o}$	
p_1	$\left(\frac{-d_o}{2}, 0\right)$	
p_2	$\left(0, \frac{-d_o}{2}\right)$	
p_3	$\left(\frac{d_o}{2}, 0\right)$	
p_4	$\left(0, \frac{d_o}{2}\right)$	
r_e	$\frac{d_o}{2}$	

SQUARE SECTION

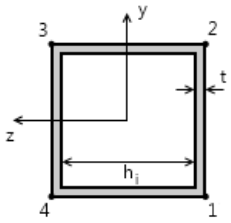


Figure 3-76: Geometry of a square shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-21: SQUARE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$4(h_i t + t^2)$	
I_{zz}	$\frac{t h_i^3 + t^3(h_i + 2t)}{6} + \frac{t(h_i + 2t)(h_i + t)^2}{2}$	
e_z	0	
μ_y	$\frac{((h_i + 2t)^3 - h_i^3)A}{16 t I_{zz}}$	
κ_y	$\frac{2(h_i + 2t)t_z}{A}$	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	
κ_z	κ_y	
J	$\frac{2t(h_i + t)^4}{h_i + t}$	Thin-walled approximation
W_t	$2t(h_i + t)^2$	Thin-walled approximation

TABLE 3-21: SQUARE SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
p_1	$\left(\frac{-(h_i+2t)}{2}, \frac{-(h_i+2t)}{2}\right)$	
p_2	$\left(\frac{(h_i+2t)}{2}, \frac{-(h_i+2t)}{2}\right)$	
p_3	$\left(\frac{(h_i+2t)}{2}, \frac{(h_i+2t)}{2}\right)$	
p_4	$\left(\frac{-(h_i+2t)}{2}, \frac{(h_i+2t)}{2}\right)$	
r_e	$\frac{h_i}{2}$	

RECTANGULAR SECTION

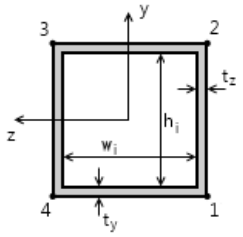


Figure 3-77: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 3-22: RECTANGULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
A	$2(h_i t_z + w_i t_y) + 4 t_y t_z$	
I_{zz}	$\frac{t_z h_i^3 + t_y^3(w_i + 2t_z)}{6} + \frac{t_y(w_i + 2t_z)(h_i + t_y)^2}{2}$	
e_z	0	

TABLE 3-22: RECTANGULAR SECTION CONSTANTS.

PROPERTY	EXPRESSION	REMARKS
μ_y	$\frac{((h_i + 2t_y)^2(w_i + 2t_z) - h_i^2 w_i)A}{16 t_z I_{zz}}$	
κ_y	$\frac{2(h_i + 2t_y)t_z}{A}$	
I_{yy}	$\frac{t_y w_i^3 + t_z^3(h_i + 2t_y)}{6} + \frac{t_z(h_i + 2t_y)(w_i + t_z)^2}{2}$	
e_y	0	
μ_z	$\frac{((w_i + 2t_z)^2(h_i + 2t_y) - w_i^2 h_i)A}{16 t_y I_{yy}}$	
κ_z	$\frac{2(w_i + 2t_z)t_y}{A}$	
J	$\frac{2(h_i + t_y)^2(w_i + t_z)^2}{\frac{h_i + t_y}{t_z} + \frac{w_i + t_z}{t_y}}$	Thin-walled approximation
W_t	$2(h_i + t_y)(w_i + t_z) \cdot \min(t_y, t_z)$	Thin-walled approximation
p_1	$\left(-\frac{h_i + 2t_y}{2}, -\frac{w_i + 2t_z}{2}\right)$	
p_2	$\left(\frac{h_i + 2t_y}{2}, -\frac{w_i + 2t_z}{2}\right)$	
p_3	$\left(\frac{h_i + 2t_y}{2}, \frac{w_i + 2t_z}{2}\right)$	
p_4	$\left(-\frac{h_i + 2t_y}{2}, \frac{w_i + 2t_z}{2}\right)$	
r_e	$\frac{h_i + 2t_y + w_i + 2t_z}{4}$	

Theory for the Phase Field in Solids Interface

The Phase Field in Solids interface solves an Allen–Cahn-type of a reaction-diffusion equation. This equation is commonly used for regularized modeling of the evolution of sharp interfaces such as cracks, damage, and grain boundaries in solids. The equation is based on the phase-field theory of fracture presented in Ref. 172. The central idea relies on approximating the set Γ that represents sharp interfaces by a regularized crack functional

$$\Gamma_l(\phi) = \int_\Omega \gamma(\phi, \nabla\phi)\mathrm{d}V \tag{3-280}$$

Here, the so-called *crack surface density function* $\gamma(\phi, \nabla\phi)$ is a function of the phase field ϕ and its gradient $\nabla\phi$, which is defined in the whole domain Ω instead of only on the crack surfaces. The crack surface density function γ also depends on the internal length scale l_{int} that controls the regularization, which means that $\Gamma_l \to \Gamma$ for $l_{\text{int}} \to 0$.

The evolution of the functional is written (Ref. 173)

$$\frac{\partial \Gamma_l}{\partial t} = S_l - R_l \geq 0 \tag{3-281}$$

where S_l is a driving source term, R_l is a viscous resistance term, and the inequality indicates the irreversibility of processes like damage and fracture. For a general crack surface density function with source and viscous resistance terms of the form

$$S_l = \frac{1}{l_{\text{int}}}\int_\Omega f\dot{\phi}\mathrm{d}V \text{ and } R_l = \frac{1}{l_{\text{int}}}\int_\Omega \tau\dot{\phi}^2\mathrm{d}V \tag{3-282}$$

the local form of Equation 3-281 can be derived as

$$\tau\dot{\phi} + l_{\text{int}}\left[\frac{\partial\gamma}{\partial\phi} - \nabla\cdot\left(\frac{\partial\gamma}{\partial(\nabla\phi)}\right)\right] = f \tag{3-283}$$

Herein, τ is a viscous regularization time constant and f is a local source term. In the derivation, homogeneous Neumann conditions are assumed on the exterior boundaries of the domain, that is,

$$\frac{\partial \gamma}{\partial(\nabla\phi)} \cdot \mathbf{N} = 0 \tag{3-284}$$

where $\mathbf{N}$ is the normal to the undeformed surface.

A common form for the crack surface density function is isotropic and quadratic in the phase field and its gradient (for example, Refs. 172–174), which is often referred to as the AT2 (Ambrosio–Tortorelli type 2) model. The crack surface density function reads

$$\gamma(\phi, \nabla\phi) = \frac{1}{2}\left[\frac{1}{l_{\text{int}}}\phi^2 + l_{\text{int}}\nabla\phi \cdot \nabla\phi\right] \tag{3-285}$$

By inserting Equation 3-285 into Equation 3-283, the strong form of the phase field equation reads

$$\tau\dot{\phi} + \phi - l_{\text{int}}^2\nabla \cdot (\nabla\phi) = f \tag{3-286}$$

The weak form of Equation 3-286 is the default equation solved in the Phase Field in Solids interface.

Alternative phase field models of fracture have considered a crack surface density function that deviate from the local quadratic term ϕ^2 and/or include anisotropy in the form of a structure tensor $\mathbf{D}$ in the nonlocal term (see Ref. 174 for a review). In the Phase Field in Solids interface, both the AT1 model

$$\gamma(\phi, \nabla\phi) = \frac{3}{8l_{\text{int}}}[\phi + l_{\text{int}}^2\nabla\phi \cdot \nabla\phi]$$

and the PF–CZM (phase-field regularized cohesive zone model)

$$\gamma(\phi, \nabla\phi) = \frac{1}{\pi l_{\text{int}}}[(2\phi - \phi^2) + l_{\text{int}}^2\nabla\phi \cdot \nabla\phi]$$

are available in addition to the standard AT2 formulation.

It is also possible to specify the phase field equation based on a generalization of Equation 3-286, which also includes an extension to multiple phase fields ϕ_k:

$$\tau_k\dot{\phi}_k + \frac{\partial Q}{\partial \phi_k} - l_{\text{int},k}^2\nabla \cdot (\mathbf{D}_k\nabla\phi_k) = f_k \tag{3-287}$$

Here, the potential function $Q(\phi_1, \phi_2, \ldots, \phi_n)$ has been introduced. Equation 3-286 is recovered for the quadratic potential

$$Q = \frac{1}{2}\sum_{k=1}^{n} \phi_k^2 \tag{3-288}$$

and the structure tensor $\mathbf{D}_k = \mathbf{I}$, which corresponds to the so-called AT2 phase-field model. Note that, in absence of a source term f, the quadratic form of the AT2 model has the advantage of admitting the trivial solution $\phi = 0$ as well as guaranteeing that the solution is bounded, $\phi \in [0, 1]$ (Ref. 172). If required, bounds on the phase field variable can be added in terms of weak inequality constraints; see Bounds in The Phase Field in Solids Interface.

Theory for the Transport in Solids Interface

The Transport Equation

The Transport in Solids interface solves the diffusion equation in solid materials. The continuity equation is formulated in the material frame (reference or undeformed configuration) for each species c_i

$$\frac{\partial c_i}{\partial t} + \nabla \cdot \Gamma_i = S_i \tag{3-289}$$

$$\Gamma_i = -D_i \nabla c_i \tag{3-290}$$

where Γ_i is the mass flux and the gradient ∇ is computed using material coordinates. When **Concentration** is used as unit for c_i (SI unit: $\mathrm{mol/m^3}$), the flux can also be expressed in terms of the gradient of chemical potential:

$$\Gamma_i = -M_i \nabla \mu_i \tag{3-291}$$

where

- c_i is the species that diffuses in the solid. It can have different units based on the user input (SI unit: $\mathrm{kg/m^3}$, $\mathrm{mol/m^3}$, or $\mathrm{1/m^3}$)
- Γ_i is the flux in the material frame (SI unit: $\mathrm{kg/(m^2 s)}$, $\mathrm{mol/(m^2 s)}$, or $\mathrm{1/(m^2 s)}$)
- S_i contains additional mass sources (SI unit: $\mathrm{kg/(m^3 s)}$, $\mathrm{mol/(m^3 s)}$, or $\mathrm{1/(m^3 s)}$)
- D_i is the diffusion tensor (SI unit: $\mathrm{m^2/s}$)
- M_i is the mobility (SI unit: $\mathrm{s \cdot mol^2/kg \cdot m^3}$)
- μ_i is the chemical potential (SI unit: J/mol)

Additional fluxes due to Electromigration, Stress Migration, and Thermophoresis can be added to Equation 3-289.

Diffusion

The diffusion coefficient in solids can be given in the reference configuration (undeformed configuration, or material frame) or in the deformed configuration (current configuration, or spatial frame). The relation between these reads

$$D = JF^{-1}dF^{-T}$$

here, D is the diffusion coefficient in the reference configuration, and d represents the diffusion coefficient in the deformed configuration. The variables F and J are the deformation gradient and its determinant.

A constant diffusion coefficient D given in material frame implies that its corresponding value in spatial frame, d, depends on the deformation

$$d = J^{-1}FDF^{T}$$

The frame in which the diffusion coefficient is given can be specified in the **Advanced** section. From the **Frame** list select **Reference configuration** or **Deformed configuration** to specify a material or spatial diffusion.

Mobility

The *mobility* M can be related to the diffusion coefficient D by comparing the definitions in Equation 3-290 and Equation 3-294:

$$\Gamma = -M\nabla\mu = -M\frac{\partial\mu}{\partial c}\nabla c = -D\nabla c$$

thus the diffusion coefficient is related to the mobility by

$$D = M\frac{\partial\mu}{\partial c}$$

For transport of dilute species, there is a linear relationship between the diffusivity and the absolute temperature:

$$D = \eta_c RT$$

where η_c is the mobility coefficient (SI unit: s mol/kg), T is the absolute temperature, and R is the gas constant.

The mobility tensor M is then proportional to the concentration c and the mobility coefficient η_c:

$$M = \eta_c c$$

so the linear mobility tensor M is related to the diffusivity as

$$M = \eta_c c = \frac{D}{RT} c$$

For a quadratic dependency on the concentration, it is written as

$$M = \eta_c c\left(1 - \frac{c}{c_{\max}}\right) = \frac{D}{RT} c\left(1 - \frac{c}{c_{\max}}\right)$$

where $c_{\max}$ is the concentration level at which there is no diffusion.

For **User defined**, it is possible enter a value or expression for the mobility tensor $\boldsymbol{M}$.

Mass Flux

Domain fluxes are vectors that can be expressed in the material frame or the spatial frame. These are related as

$$\Gamma = JF^{-1}\gamma$$

Here, Γ is the flux in the reference (undeformed) configuration, γ is the flux in the deformed (current) configuration, $\boldsymbol{F}$ is the deformation gradient, and J is its determinant.

Normal Flux

When specifying normal boundary fluxes as boundary conditions, it is assumed that the normal flux follows the deformed boundary.

The normal flux to the reference (undeformed) configuration is defined as

$$-\Gamma \cdot \mathbf{N} = \Gamma_n$$

where the normal flux to the deformed configuration is defined as

$$-\gamma \cdot \mathbf{n} = \gamma_n$$

Here, $\mathbf{N}$ is the normal to the undeformed boundary, $\mathbf{n}$ is the normal to the deformed boundary.

The normal fluxes in the material frame (undeformed configuration) and spatial frame (deformed configuration) are related to as

$$\Gamma_n = J_A \gamma_n$$

where the variable J_A is the ratio between the deformed area and the reference area.

By default, you specify the normal flux γ_n to the deformed configuration (spatial frame), but it is also possible to specify the normal flux in the reference configuration (material frame), Γ_n. The frame in which the normal flux is given can be specify in the **Advanced** section.

Mass Balance

The mass balance equation states that the total variation of mass, m, in a domain, V, is caused by the mass entering or exiting its surrounding boundary, and mass sources or sinks inside the domain

$$\frac{\partial m_i}{\partial t} = -\int_S \Gamma_i \cdot \mathbf{N} dS + \int_V S_i dV \tag{3-292}$$

The time variation of the mass in the domain given in the reference configuration can be expressed as follows

$$\frac{\partial m_i}{\partial t} = \frac{\partial}{\partial t}\int_V dm_i = \frac{\partial}{\partial t}\int_V \rho_0 dV$$

where ρ_0 is the density in the undeformed configuration. It is related to the current density through the volume ratio

$$\rho_0 = J\rho$$

After using the divergence theorem for the source term, Equation 3-292 reads

$$\frac{\partial \rho_0}{\partial t} = -\nabla \cdot \Gamma_i + S_i$$

The flux vector Γ contains the Fickian diffusion plus external fluxes

$$\Gamma_i = \Gamma_{\text{dif}} + \Gamma_{\text{ext}}$$

External Flux

The **External Flux** node ca be used to add an additional flux contribution ($\Gamma_{\text{ext},\,i}$) which is not part of the constitutive relation. Multiple subnodes can be added. The transport equation is modified as follows,

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (\Gamma_i + \Gamma_{\text{ext},i}) = S_i$$

Chemical Potential

The *chemical potential* μ is the driving force for transport processes and it is related to the concentration c through the *free energy density*, ψ, as follows

$$\mu = \frac{\partial \psi}{\partial c}$$

The free energy density can have contributions from different sources like the elastic energy due to deformation, or the mixing energy of a solvent in the solid

$$\psi = \psi_{\text{ch}} + \psi_{\text{el}} + \dots$$

The chemical potential is decomposed as follows

$$\mu = \frac{\partial \psi_{\text{ch}}}{\partial c} + \frac{\partial \psi_{\text{ext}}}{\partial c} = \mu_{\text{ch}} + \mu_{\text{ext}} \tag{3-293}$$

where μ_{ext} contains all the contributions not related to the diffusion process. Note that the free energy density, ψ, has to be function of the concentration c to define the chemical potential.

For instance, the Flory–Huggins free energy often used to describe diffusion in polymeric gel reads

$$\psi_{\text{ch}}(c) = -\frac{N_A k_B T}{V_{\text{m}}}\left[V_m c \log\left(1 + \frac{1}{V_{\text{m}} c}\right) + \frac{\chi}{1 + V_{\text{m}} c}\right]$$

It is possible to define the diffusive flux as a function of the chemical potential gradient

$$\Gamma = -M \nabla \mu \tag{3-294}$$

where M is the *mobility*, that can be related to the diffusion coefficient D by comparing the definitions in Equation 3-290 and Equation 3-294

$$\Gamma = -M \nabla \mu = -M \frac{\partial \mu}{\partial c} \nabla c = -D \nabla c$$

thus the diffusion coefficient is related to the mobility by

$$D = M\frac{\partial \mu}{\partial c}$$

The mobility tensor M is the proportionality factor between the gradient of the chemical potential and the diffusive flux, and it is often a function of the concentration

$$\Gamma = -M(c)\nabla\mu$$

For a linear mobility, it is written as

$$M = \eta_c c$$

where η_c is the mobility coefficient (SI unit: s·mol/kg).

For a quadratic dependency on the concentration, it is written as

$$M = \eta_c c\left(1 - \frac{c}{c_{max}}\right)$$

where c_{max} is the concentration level at which there is no diffusive flux.

It is also possible to enter a value or expression for the mobility tensor M (SI unit: $s{\cdot}mol^2/(kg{\cdot}m^3)$).

Electromigration

The free energy density can have a contribution from the electric energy. Therefore, the chemical potential in Equation 3-293 can depend on the electric potential V as

$$\mu_{em} = zFV$$

where z is the charge number of the transported species, and F is Faraday constant. The flux due to electromigration is then defined as a function of the chemical potential

$$\Gamma_{em} = -M\nabla\mu_{em} = -M\frac{\partial \mu_{em}}{\partial V}\nabla V = -MzF\nabla V$$

with the mobility M and the electric field $E = -\nabla V$.

For linear mobility, M is related to the diffusion coefficient D as

$$M = \frac{D}{RT}c$$

so the electromigration flux reads

$$\Gamma_{\mathrm{em}} = -\frac{D}{RT} czF\nabla V$$

Replacing Faraday constant with the elementary charge and Avogadro number, $F = eN_{\mathrm{A}}$, and the gas constant in terms of Boltzmann constant, $R = k_{\mathrm{B}}N_{\mathrm{A}}$, gives the electromigration flux

$$\Gamma_{\mathrm{em}} = -\frac{D}{k_{\mathrm{B}}T} cze\nabla V$$

Stress Migration

The free energy density can have a contribution from the elastic energy. Therefore, the chemical potential in Equation 3-293 can depend on the mean stress σ_{m},

$$\mu_{\mathrm{sm}} = -\sigma_{\mathrm{m}} V_{\mathrm{m}}$$

with the partial molar volume V_{m}. The flux from stress migration is defined as a function of the chemical potential,

$$\Gamma_{\mathrm{sm}} = -M\nabla\mu_{\mathrm{sm}} = M\nabla(\sigma_{\mathrm{m}} V_{\mathrm{m}})$$

where V_{m} corresponds to the partial molar volume.

Thermophoresis

The free energy density can have a contribution from the thermal energy. Therefore, the chemical potential in Equation 3-293 can have the contribution

$$\mu_{\mathrm{th}} = RTS_{\mathrm{T}}$$

where R is the universal gas constant, T is the temperature, and S_{T} is Soret coefficient. The flux from thermophoresis is then defined as a function of the chemical potential,

$$\Gamma_{\mathrm{th}} = -M\nabla\mu_{\mathrm{th}} = -M\frac{\partial\mu_{\mathrm{th}}}{\partial T}\nabla T = -MRS_{\mathrm{T}}\nabla T$$

For linear mobility, M is related to the diffusion coefficient D as

$$M = \frac{D}{RT} c$$

so the thermophoresis flux reads

$$\Gamma_{\text{th}} = -\frac{D}{T} c S_{\text{T}} \nabla T$$

It s also possible to define the thermal diffusion coefficient $D_{\text{T}} = DS_{\text{T}}$, in which case the thermophoresis flux reads

$$\Gamma_{\text{th}} = -\frac{D_{\text{T}}}{T} c \nabla T$$

Electrode Surface Coupling

Use the **Electrode Surface Coupling** node to specify the mass flux into or out of the domain based on current densities specified in an Electrochemistry. The following generic electrochemical reaction is solved

$$\sum_{\text{ox}} |\nu_{\text{ox}}| \text{Ox} + ne^- \Leftrightarrow \sum_{\text{red}} \nu_{\text{red}} \text{Red}$$

The normal mass flux on the surface is defined as

$$-\mathbf{n} \cdot \gamma_i = \sum_m r_{i,m}$$

where the mth reaction is defined as

$$r_{i,m} = \frac{-\nu_{i,m} i_{\text{loc}}}{nF}$$

Here, F is Faraday constant, n is the number of participating electrons in the reaction, $\nu_{i,m}$ are the stoichiometric coefficients, and i_{loc} is the normal current density of the electrode reaction.

The following convention is used: If the transported species is the reduced species, the stoichiometric coefficient ν_{red} is set to a positive value. Otherwise, the stoichiometric coefficient ν_{ox} is set to a negative value.

The **Electrode Surface Coupling** node is available in a limited set of add-on products. See www.comsol.com/products/specifications/ for details.

See the Theory for the Surface Reactions Interface section in the *Chemical Species Transport Interfaces* chapter.

References

REFERENCES FOR NONLINEAR STRUCTURAL MATERIALS

1. G.A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, Wiley, 2000.

2. R.S. Rivlin and D.W. Saunders, "Large Elastic Deformations of Isotropic Materials. VII. Experiments on the Deformation of Rubber," *Phil. Trans. R. Soc. Lond. A*, vol. 243, no. 865, pp. 251–288, 1951.

3. E.M. Arruda and M.C. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," *J. Mech. Phys. Solids*, vol. 41, pp. 389–412, 1993.

4. J.C. Simo and T.J.R. Hughes, *Computational Inelasticity*, Springer, 1998.

5. J. Lubliner, *Plasticity Theory*, Dover, 2008.

6. K. Grote and E. Antonsson, *Springer Handbook of Mechanical Engineering*, Springer, 2009.

7. R. Hertzberg, *Deformation and Fracture Mechanics of Engineering Materials*, John Wiley & Sons, 1996.

8. F. Garofalo, "An Empirical Relation Defining the Stress Dependence of Minimum Creep Rate in Metals," *Trans AIME*, vol. 227, no. 351, 1963.

9. S. Brown, K. Kim, and L. Anand, "An Internal Variable Constitutive Model for Hot Working of Metals," *Int. J. Plasticity*, vol. 5, pp. 95–130, 1989.

10. L. Anand and S. Narayan, "An Elastic-Viscoplastic Model for Lithium". *J. Electrochem. Soc.*, 166, A1092–A1095, 2019.

11. J. Simo and K. Pister, "Remarks on rate constitutive equations for finite deformation problems: computational implications," *Comput. Methods Appl. Mech. Eng.*, vol. 46, no. 2, pp. 201–215, 1984.

12. J. Simo, R. Taylor, and K. Pister, "Variational and projection methods for the volume constraint in finite deformation elasto-plasticity," *Comput. Methods Appl. Mech. Eng.*, vol. 51, pp. 177–208, 1985.

13. S. Hartmann and P. Neff, "Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility," *Int. J. Solids Struct.*, vol. 40, pp. 2767–2791, 2003.

14. M. Attard, "Finite Strain — Isotropic Hyperelasticity," *Int. J. Solids Struct.*, vol. 40, pp. 4353–4378, 2003.

15. C. Miehe, "Aspects of the formulation and finite element implementation of large strain isotropic elasticity," *Int. J. Numer. Meth. Engng.*, vol. 37, pp. 1981–2004, 1994.

16. J. Simo, R. Taylor, "Quasi-incompressible finite elasticity in principal stretches. continuum basis and numerical algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 85, pp. 273–310, 1991.

17. I. Babuska and M. Suri, "On Locking and Robustness in the Finite Element Method," *SIAM Journal on Numerical Analysis*, vol. 29, pp. 1261–1293, 1992.

18. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, 2009.

19. C. Kadapa and M. Hossain, "A linearized consistent mixed displacement-pressure formulation for hyperelasticity," *Mech. Adv. Mater. Struct.*, vol. 29, pp. 267–284, 2022.

20. A. Gent, "A new Constitutive Relation for Rubbers," *Rubber Chem. Technol.*, vol. 69, pp. 59–61, 1996.

21. C. Horgan and G. Saccomandi, "Constitutive Models for Compressible Nonlinearly Elastic Materials with Limiting Chain Extensibility," *J. Elasticity*, vol. 7, pp. 123–138, 2004.

22. H. Ambacher, H. Enderle, H. Kilian, A. Sauter, "Relaxation in permanent networks". In: Pietralla M., Pechhold W. (eds) *Relaxation in Polymers. Progress in Colloid & Polymer Science*, vol. 80, pp. 209–220, Steinkopff, 1989.

23. M. Kaliske and H. Rothert, "On the finite element implementation of rubber-like materials at finite strains," *Eng. Comput.*, vol. 14, pp. 216–232, 1997.

24. G. Marckmann and E. Verron, "Comparison of hyperelastic models for rubber-like materials" *Rubber Chem. Technol.*, vol. 79, pp. 835–858, 2006.

25. B. Storakers, "On Material Representation and Constitutive Branching in Finite Compressible Elasticity," *J. Mech. Phys. Solids*, vol. 34, pp. 125–145, 1986.

26. P. Blatz, W. Ko , "Application of Finite Elastic Theory to the Deformation of Rubbery Materials," *Tran. Soc. Rheol.*, vol. 6, pp. 223–251, 1962.

27. Y. Gao, "Large Deformation Field Near a Crack Tip in a Rubber-like Material," *Theor. Appl. Fract. Mec.*, vol. 26, pp. 155–162, 1997.

28. R.L. Taylor, "Thermomechanical analysis of viscoelastic solids," *Int. J. Numer. Meth.*, vol. 2, pp. 45–59, 1970.

29. G. Holzapfel, T. Gasser, and R. Ogden, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models," *J. Elasticity*, vol. 61, pp. 1–48, 2000.

30. T. Gasser, R. Ogden, and G. Holzapfel, "Hyperelastic modelling of arterial layers with distributed collagen fibre orientations," *J. R. Soc. Interface*, vol. 3, pp. 15–35, 2006.

31. J. Helfenstein and others, "On non-physical response in models for fiber-reinforced hyperelastic materials," *Int. J. Solids Struct.*, vol. 47, pp. 2056–2061, 2010.

32. D. Nolan and others, "A robust anisotropic hyperelastic formulation for the modelling of soft tissue," *J. Mech. Behav. Biomed. Mater.*, vol. 39, pp. 48–60, 2014.

33. M. Kaliske and G. Heinrich, "An Extended Tube Model for Rubber Elasticity: Statistical-Mechanical Theory and Finite Element Implementation," *Rubber Chem. Technol.*, vol. 72, pp. 602–632, 1999.

34. R. Behnke and M. Kaliske, "The Extended Non-affine Tube Model for Crosslinked Polymer Networks: Physical Basics, Implementation, and Application to Thermomechanical Finite Element Analyses," in *Designing of Elastomer Nanocomposites: From Theory to Applications*, Springer, 2017.

35. J. Bergström, *Mechanics of Solid Polymers: Theory and Computational Modeling*, Elsevier, 2015.

36. J. Bergström and M. Boyce, "Large strain time-dependent behavior of filled elastomers" *Mechanics of Materials*, vol. 32, pp. 627–644, 2000.

37. J. Bergström and J. Bischoff, "An advanced thermomechanical constitutive model for UHMWPE," *The International Journal of Structural Changes in Solids*, vol. 2 pp. 31–39, 2010.

38. A. Souza and others, "Three-dimensional model for solids undergoing stress-induced phase transformations," *European J. Mech. A Solids*, vol. 17, pp. 789–806, 1988.

39. F. Auricchio and L. Petrini, "A three-dimensional model describing stress-temperature induced solid phase transformations. Part I: solution algorithm and boundary value problems," *Int. J. Numer. Meth. Eng.*, vol. 61, pp. 807–836, 2004.

40. F. Auricchio and others, "SMA numerical modeling versus experimental results: Parameter identification and model prediction capabilities," *J. Mater. Eng. Perform.*, vol. 18, pp. 649–654, 2009.

41. D. Lagoudas (Ed.) *Shape Memory Alloys: Modeling and Engineering Applications*, Springer, 2008.

42. D. Lagoudas and others, "Constitutive model for the numerical analysis of phase transformation in polycrystalline shape memory alloys," *Int. J. Plast.*, vol. 32–33, pp. 155–183, 2012.

43. L. Mullins, "Effect of stretching on the properties of rubber," *J. Rubber Res.*, vol. 16, pp. 275–289, 1947.

44. L. Mullins and N.R. Tobin, "Theoretical model for the elastic behavior of filled-reinforced vulcanized rubbers," *Rubber Chem. Technol.*, vol. 30, pp. 555–571, 1957.

45. L. Mullins. "Softening of rubber by deformation," *Rubber Chem. Technol.*, vol. 42, pp. 339–362, 1969.

46. R. Ogden and D. Roxburgh, "A pseudo-elastic model for the Mullins effect in filled rubber," *Proc. R. Soc. Lond.*, vol. A 455, pp. 2861–2878, 1999.

47. A. Dorfmann and R. Ogden, "A constitutive model for the Mullins effect with permanent set in particle-reinforced rubber," *Int. J. Solids Struct.*, vol. 41, pp. 1855–1878, 2004.

48. Z. Guo and L. Sluys, "Computational modelling of the stress-softening phenomenon of rubber-like materials under cyclic loading," *Eur. J. Mech. A/Solids*, vol. 25, pp. 877–896, 2006.

49. C. Miehe, "Discontinuous and continuous damage evolution in Ogden-type large strain elastic materials," *Eur. J. Mech. A/Solids*, vol. 14, pp. 697–720, 1995.

50. R. Koeller, "Application of fractional calculus to the theory of viscoelasticity," *J. Appl. Mech.*, vol. 51. pp. 299–307, 1984.

51. M. Rezaee-Hajidehi, K. Tuma, S. Stupkiewicz, et.al., "A note on Padé approximants of tensor logarithm with application to Hencky-type hyperelasticity," *Comput. Mech.* vol. 68, pp. 619–632, 2021.

52. D. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," CMAME, vol. 99, pp. 235–394, 1992.

53. D. Flanagan T. and Belytschko. "A uniform strain hexahedron and quadrilateral with orthogonal hourglass control," Int. J. Numer. Meth. Engng., vol. 17, pp. 679–706, 1981.

REFERENCES FOR GEOMECHANICS MATERIALS

54. W.F. Chen and E. Mizuno, *Nonlinear Analysis in Soil Mechanics: Theory and Implementation (Developments in Geotechnical Engineering)*, 3rd ed., Elsevier Science, 1990.

55. D. Drucker and W. Prager, "Soil Mechanics and Plastic Analysis or Limit Design," *Q. Appl. Math.*, vol. 10, no. 2, pp. 157–165, 1952.

56. S. Dolarevic and A. Ibrahimbegovic, "A modified three-surface elasto-plastic cap model and its numerical implementation," *Comput. Struct.*, vol. 85, pp. 419–430, 2007.

57. B. Bresler and K. Pister, "Strength of Concrete Under Combined Stresses," *ACI Journal*, vol. 551, pp. 321–345, 1958.

58. H. Matsuoka and T. Nakai, "Stress-deformation and Strength Characteristics of Soil Under Three Different Principal Stresses," *Proc. JSCE*, vol. 232, 1974.

59. H. Matsuoka and T. Nakai, "Relationship Among Tresca, Mises, Mohr–Coulomb, and Matsuoka–Nakai Failure Criteria," *Soils Found.*, vol. 25, pp. 123–128, 1985.

60. H.S. Yu, *Plasticity and Geotechnics*, Springer, 2006.

61. V. Marinos, P. Marinos, and E. Hoek, "The Geological Strength Index: Applications and Limitations," *Bull. Eng. Geol. Environ.*, vol. 64, pp. 55–65, 2005.

62. J. Jaeger, N.G. Cook, and R. Zimmerman, *Fundamentals of Rock Mechanics*, 4th ed., Wiley-Blackwell, 2007.

63. G. C. Nayak and O.C. Zienkiewicz, "Convenient Form of Stress Invariants for Plasticity," *J. Struct. Div. ASCE*, vol. 98, pp. 949–954, 1972.

64. A.J. Abbo and S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr–Coulomb Yield Criterion," *Comput. Struct.*, vol. 54, pp. 427–441, 1995.

65. K.J. Willam and E.P. Warnke, "Constitutive Model for the Triaxial Behavior of Concrete," IABSE Reports of the Working Commissions, Colloquium (Bergamo): *Concrete Structures Subjected to Triaxial Stresses*, vol. 19, 1974.

66. B.H.G. Brady and E.T. Brown, *Rock Mechanics for Underground Mining*, 3rd ed., Springer, 2004.

67. H.A. Taiebat and J.P. Carter, "Flow Rule Effects in the Tresca Model," *Comput. Geotech.*, vol. 35, pp. 500–503, 2008.

68. A. Stankiewicz and others, "Gradient-enhanced Cam-Clay Model in Simulation of Strain Localization in Soil," *Foundations of Civil and Environmental Engineering*, no. 7, 2006.

69. D.M. Wood, *Soil Behaviour and Critical State Soil Mechanics*, Cambridge University Press, 2007.

70. D.M. Potts and L. Zadravkovic, *Finite Element Analysis in Geothechnical Engineering*, Thomas Telford, 1999.

71. W. Tiecheng and others, *Stress-strain Relation for Concrete Under Triaxial Loading*, 16th ASCE Engineering Mechanics Conference, 2003.

72. W.F. Chen, *Plasticity in Reinforced Concrete*, McGraw Hill, 1982.

73. N. Ottosen, "A Failure Criterion for Concrete," *J. Eng. Mech. Division, ASCE*, vol. 103, no. 4, pp. 527–535, 1977.

74. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modelling*, Elsevier, 2005.

75. J. Suebsuk, S. Horpibulsuk, and M. Liu, "Modified Structured Cam Clay: A generalised critical state model for destructured, naturally structured and artificially structured clays," *Comput. Geotech.*, vol. 37, pp. 956–968, 2010.

76. M. Lui and J.P. Carter, "A structured Cam Clay model," *Can. Geotech. J.*, vol. 39, pp. 1313–1332, 2002.

77. E. Alonso, A. Gens, and A. Josa, "A constitutive model for partially saturated soils," *Géotechnique*, vol. 40, 1990.

78. D. Pedroso and M. Farias, "Extended Barcelona Basic Model for unsaturated soil under cyclic loadings," *Comput. Geotech.*, vol. 38, no. 5, pp. 731–740, 2011.

79. T. Benz, *Small-strain stiffness of soils and its numerical consequences*, PhD Thesis, Stuttgart University, 2007.

80. T. Benz, P. Vermeer, and R. Schwab, "A Small-Strain Overlay Model," *Int. J. Numer. Anal. Meth. Geomech.*, vol. 33, pp. 25–44, 2009.

81. T. Bower, *Constitutive modelling of soils and fibre-reinforced soils*, PhD Thesis, Cardiff University, 2017.

82. P. Grassl and M. Jirásek, "Damage-plastic model for concrete failure," *Int. J. Solids Struct.,* vol. 43, pp. 7166–7196, 2006.

83. P. Grassl, D. Xenos, U. Nyström, R. Rempling, and K. Gylltoft, "CDPM2: A damage-plasticity approach to modelling the failure of concrete," *Int. J. Solids Struct.,* vol. 50, pp. 3805–3816, 2013.

84. A. Wilkins and others, "A method for smoothing multiple yield functions," *Int. J. Numer. Methods Eng.*, vol. 121, pp. 434–449, 2020.

REFERENCES FOR THE NONLINEAR ELASTIC MATERIAL THEORY

85. W. Ramberg and W.R. Osgood, "Description of stress-strain curves by three parameters," *NACA Technical Note*, no. 902, 1943.

86. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, 2009.

87. B.O. Hardin and V.P. Drnevich, "Shear modulus and damping in soils: Design equations and curves," *J. Soil Mech.*, vol. 98, no. 6, pp. 667–692, 1972.

88. J.M. Duncan and C.Y. Chang, "Nonlinear analysis of stress and strain in soils," *J. Soil Mech.*, vol. 96, no. 5, pp. 1629–1653, 1970.

89. J.M. Duncan, P. Byrne, K.S. Wong, and P. Mabry, "Strength, stress strain and bulk modulus parameters for finite element analysis of stresses and movements in soil," *Geotechnical Engineering Report: UCB/GT/80-01*, University of California, Berkeley, 1980.

90. E.T. Selig, *Soil Parameters for Design of Buried Pipelines, Pipeline Infrastructure*, B.A. Bennett, ed., ASCE, New York, pp. 99–116, 1988.

REFERENCES FOR ELASTOPLASTIC MATERIALS

91. P. Armstrong and C. Frederick, "A Mathematical Representation of the Multiaxial Bauschinger Effect," Technical Report RD/B/N731 CEGB, 1966. Reprinted in *Mater. High Temp.*, vol. 24, no. 1, pp. 1–26, 2007.

92. M. Jirasek and Z. Bazant, *Inelastic Analysis of Structures*, *Ch.20 General Elastoplastic Constitutive Models*. Wiley, 2001.

93. J. Simo and T. Hughes, *Computational Inelasticity*, Springer, 1998.

94. J. Simo, "Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory," *Comput. Methods Appl. Mech. Eng.*, vol. 99, pp. 61–112, 1992.

95. J. Lubliner, *Plasticity Theory*, Dover, 2008.

96. R. Hill, "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proc. Roy. Soc. London*, vol. 193, pp. 281–297, 1948.

97. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modeling,* Elsevier Science, 2005.

98. A Lion, "Constitutive modelling in finite thermoviscoplasticity: a physical approach based on nonlinear rheological models," *Int. J. Plast.,* 2000.

99. G. Gudehus, "A Comprehensive Constitutive Equation for Granular Materials," *Soils Found.*, vol. 36, pp. 1–12, 1996.

100. R. Lagioia and A. Panteghini, "On the existence of a unique class of yield and failure criteria comprising Tresca, von Mises, Drucker–Prager, Mohr–Coulomb, Galileo–Rankine, Matsuoka–Nakai and Lade–Duncan," *Proc. R. Soc. A,* vol. 472: 20150713, 2016.

101. S. Shima and M. Oyane, "Plasticity theory for porous metals," *Int. J. Mech. Sci.*, vol. 18, pp. 285–291, 1976.

102. A. Gurson, "Continuum theory of ductile rupture by void nucleation and growth: Part I – Yield criteria and flow rules for porous ductile media," *J. Eng. Mater. Technol.*, vol. 99, pp. 2–15, 1977.

103. V. Tvergaard and A. Needleman, "Analysis of the cup-cone fracture in a round tensile bar," *Acta Metallurgica*, vol. 32, pp. 157–169, 1984.

104. A. Needleman and V. Tvergaard, "An analysis of ductile rupture in notched bars," *J. Mech. Phys. Solids*, vol. 32, pp. 461–490, 1984.

105. K. Nahshon and Z. Xue, "A modified Gurson model and its application to punch-out experiments," *Eng. Fract. Mech.*, vol. 76, pp. 997–1009, 2009.

106. A.E. Huespe, A. Needleman, and J. Oliver, "A finite strain, finite band method for modeling ductile fracture," *Int. J. Plast.,* vol. 28, pp. 53–69, 2012.

107. N. Fleck, L. Kuhn, and R. McMeeking, "Yielding of metal powder bonded by isolated contacts," *J. Mech. Phys. Solids*, vol. 40, pp. 1139–1162, 1992.

108. P. Redanz, "Numerical modelling of the powder compaction of a cup," *Eur. J. Mech. - A/Solids*, vol. 18, pp. 399–413, 1999.

109. J. Cedergren, N. Sorensen, and A. Bergmark, "Three-dimensional analysis of compaction of metal powder," *Mech. Mater.*, vol. 34, pp. 43–59, 2004.

110. S. Tsai and E. Wu, "A general theory of strength for anisotropic materials," *J. Compos. Mater.*, vol. 5, pp. 58–80, 1971.

111. E. Martínez-Pañeda and others, "Griffith-based analysis of crack initiation location in a Brazilian test," *Int. J. Rock Mech. Min. Sci.*, vol. 159, p. 105227, 2022.

112. P. Hopkins, "Benchmarks for Membrane and Bending Analysis of Laminated Shells. Part 2: Strength Analysis," NAFEMS Ltd, 2005.

113. Z. Hashin, "Failure Criteria for Unidirectional Fiber Composites," *J. Appl. Mech.*, vol. 47, p. 329, 1980.

114. A.J. Sobey, J.I.R. Blake, and R.A. Shenoi, "Implications of failure criteria choices on the rapid concept design of composite grillage structures using multiobjective optimisation," *Struct. Multidiscipl. Optim.*, vol. 47, pp. 735–747, 2013.

115. C.G. Davila, P.P. Camanho, and C.A. Rose. "Failure Criteria for FRP Laminates," *J. Compos. Mater.*, 39, 2005.

116. R.A.B Engelen, M.G.D. Geers, and F.P.T. Baaijens, "Nonlocal implicit gradient-enhanced elasto-plasticity for modelling of softening behaviour," *Int. J. Plast.*, vol. 19, no. 4, pp. 403–433, 2003.

117. S. Forest, "Micromorphic approach for gradient elasticity, viscoplasticity, and damage," *J. Eng. Mech.*, vol. 135, no. 3, pp. 117–131, 2009.

118. D.P Adhikary, C.T. Jayasundara, R.K. Podgorney, and A.H. Wilkins "A robust return-map algorithm for general multisurface plasticity," *Int. J. Numer. Eng.*, vol. 109, no. 2, pp. 218-234, 2016.

119. V.S Deshpande, and N.A. Fleck, "Isotropic constitutive models for metallic foams," *J. Mech. Phys. Solids*, vol. 49, no. 6-7, pp. 1253-1283, 2000.

REFERENCES FOR DAMAGE MODELS

120. R. de Borst, M. Crisfield, J. Remmers, and C. Verhoosel, *Non-linear Finite Element Analyses of Solids and Structures*, John Wiley & Sons, 2012.

121. M. Jirásek, "Damage and smeared crack models," pp. 1–49 in *Numerical Modeling of Concrete Cracking*, G. Hofstetter and G. Meschke eds., Springer, 2011.

122. J. Mazars, "A description of micro and macroscale damage of concrete structures," *Eng. Fract. Mech.*, vol. 25, pp. 729–737, 1986.

123. Z. Bazant and B. Oh, "Crack band theory for fracture in concrete," *Mater. Struct.*, vol. 16, pp. 155–177, 1983.

124. M. Jirásek and M. Bauer, "Numerical aspects of the crack band approach," *Comput. Struct.*, vol. 110, pp. 60–78, 2012.

125. R. Peerlings and others, "Gradient enhanced damage for quasi-brittle materials," *Int. J. Numer. Meth. Eng.*, vol. 39, pp. 3391–3403, 1996.

126. C. Miehe, M. Hofhacker, and F. Welschinger, "A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits," *Comput. Methods Appl. Mech. Eng.*, vol. 199, pp. 2765–2778, 2010.

127. F. Hildebrand, C. Miehe, "Comparison of Two Bulk Energy Approaches for the Phase field Modeling of Two-variant Martensitic Laminate Microstructure," T*ech. Mec.* vol. 32, pp. 3–20, 2012.

128. C. Miehe, L.S. Schänzel, and H. Ulmer, "Phase field modeling of fracture in multi-physics system. Part 1. Balance of crack surface and failure criteria for brittle crack propagation in thermo-elastic solids," *Comput. Methods Appl. Mech. Eng.*, vol. 294, pp. 449–485, 2014.

129. J.-Y. Wu, V.P. Nguyen, C.T. Nguyen, D. Sutula, S. Sinaie, and S. Bordas, "Phase-field modelling of fracture," *Advances in Applied Mechanics*, vol. 53, pp. 1–183, 2020.

130. M.J Borden, *Analysis of Phase-Field Models for Dynamic Brittle and Ductile Fracture*, PhD Thesis, The University of Texas at Austin, 2012.

REFERENCES FOR WAVE ATTENUATION

131. R. Newnham, *Properties of Materials. Anisotropy, Symmetry, Structure.* Oxford University Press, 2005.

132. K. Ono, "Review on Structural Health Evaluation with Acoustic Emission," *Appl. Sci.*, vol. 8, p. 958, 2018.

REFERENCES FOR PIEZOELECTRICITY

133. R. Holland and E.P. EerNisse, *Design of Resonant Piezoelectric Devices*, Research Monograph no. 56, The MIT. Press, 1969.

134. T. Ikeda, *Fundamentals of Piezoelectricity*, Oxford University Press, 1990.

135. A.V. Mezheritsky, "Elastic, Dielectric, and Piezoelectric Losses in Piezoceramics: How it Works all Together," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 6, 2004.

136. K. Uchino and S. Hirose, "Loss Mechanisms in Piezoelectrics: How to Measure Different Losses Separately," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 1, pp. 307–321, 2001.

137. P.C.Y. Lee, N.H. Liu, and A. Ballato, "Thickness Vibrations of a Piezoelectric Plate with Dissipation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 1, 2004.

138. P.C.Y. Lee and N.H. Liu, "Plane Harmonic Waves in an Infinite Piezoelectric Plate with Dissipation," *Frequency Control Symposium and PDA Exhibition*, IEEE International, pp. 162–169, 2002.

139. C.A. Balanis, "Electrical Properties of Matter," *Advanced Engineering Electromagnetics*, John Wiley & Sons, 1989.

140. J. Yang, *An Introduction to the Theory of Piezoelectricity*, Springer Science and Business Media, 2005.

REFERENCES FOR MAGNETOMECHANICS

141. J.A. Stratton, *Electromagnetic Theory*, Cambridge, MA, 1941.

142. L.D. Landau and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Pergamon Press, pp. 69–73, 1960.

143. D.J. Griffiths, "Resource Letter EM-1: Electromagnetic Momentum," *Am. J. Phys.*, vol. 80, pp. 7–18, 2012.

144. L. Dorfmann, R.W. Ogden, *Mechanics and Electrodynamics of Electroelastic and Magnetoelastic Interactions*, Springer, 2014.

REFERENCES FOR MAGNETOSTRICTION

145. Chikazumi, *Physics of Ferromagnetism*, Oxford University Press, New York, 1997.

146. H. Kronmüller, "General Micromagnetic Theory," in *Handbook of Magnetism and Advanced Magnetic Materials*, edited by H. Kronmüller and S. Parkin, Vol. 2: *Micromagnetism*, John Wiley & Sons, Chichester, 2007.

147. X.E. Liu and X.J. Zheng, "A Nonlinear Constitutive Model for Magnetostrictive Materials," *Acta Mech. Sinica*, vol. 21, pp. 278–285, 2005.

148. D.C. Jiles, *Introduction to Magnetism and Magnetic Materials*, 2nd ed., Chapman & Hall, London, 1998.

149. M.J. Dapino, "Nonlinear and Hysteretic Magnetomechanical Model for Magnetostrictive Transducers," PhD Dissertation, Iowa State University, 1999.

150. C.H. Sherman and J.L. Butler, Appendix A.7, p. 555 in *Transducers and Arrays for Underwater Sound*, Springer, New York, 2007.

REFERENCES FOR ELECTROSTRICTION

151. J.A. Stratton, *Electromagnetic Theory*, Cambridge, MA, 1941.

152. L.D. Landau and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Pergamon Press, pp. 69–73, 1960.

153. R.E. Newnham, V. Sundar, R. Yimnirun, J. Su, and Q.M. Zhang, "Electrostriction: Nonlinear Electromechanical Coupling in Solid Dielectrics," *J. Phys. Chem. B*, vol. 101, pp. 10141–10150, 1997.

REFERENCES FOR FERROELASTICITY

154. F. Li, Jin, Z. Xu, and S. Zhang, "Electrostrictive effect in ferroelectrics: An alternative approach to improve piezoelectricity," *Appl. Phys. Reviews*, vol. 1, no. 1, pp. 011103-1–011103-21, 2014.

155. C. L. Horn and N. Shankar, "A finite element method for electrostrictive ceramic devices," *Int. J. Solids Struct.*, vol. 33, pp. 1757–1779, 1995.

156. R.C. Smith and Z. Ounaies. "A Domain Wall Model for Hysteresis in Piezoelectric Materials," *J. Int. Mat. Sys. Struct.*, vol. 11, no. 1, pp. 62–79, 2000.

REFERENCES FOR BOUNDARY CONDITIONS

157. M. Cohen and P.C. Jennings, "Silent Boundary Methods for Transient Analysis," *Computational Methods for Transient Analysis*, vol 1 (editors T. Belytschko and T.J.R. Hughes), North-Holland, 1983.

158. B. Lalanne and M. Touratier, "Aeroelastic Vibrations and Stability in Cyclic Symmetric Domains," *Int. J. Rotating Mach.*, vol. 6, no. 6, pp. 445–452, 2000.

159. Y.C. Fung, *Foundations of Solid Mechanics*, Prentice-Hall, 1965, page 99.

160. Q. Hu, F. Chouly, P. Hu, G. Cheng, and S.P.A Bordas, "Skew-symmetric Nitsche's formulation in isogeometric analysis: Dirichlet and symmetry conditions, patch coupling and frictionless contact," *Comput. Methods Appl. Mech. Eng.*, vol. 341, pp. 188–220, 2018.

REFERENCES FOR CONTACT ANALYSIS AND DECOHESION

161. R. Mlika, Y. Renard, and F. Chouly, "An unbiased Nitsche's formulation of large deformation fictional contact and self-contact," *Computer Methods in Applied Mechanics and Engineering*, vol. 325, pp. 265–288, 2017

162. L. De Lorenzis, D. Fernando, and J.G. Teng, "Coupled mixed-mode cohesive zone modeling of interfacial debonding in plated beams," *Int. J. Solids Struct.*, vol. 50, pp. 2477–2494, 2013.

163. R.D.S.G. Campilho, M.F.S.F. de Moura, and J.J.M.S. Domingues, "Using a cohesive damage model to predict the tensile behavior of CFRP single-strap repairs," *Int. J. Solids Struct.*, vol. 45, no. 5, pp. 1497–1512, 2008.

164. S.T. Pinho, L. Iannucci, and P. Robinson, "Formulation and implementation of decohesion elements in an explicit finite code," *Composites Part A: Applied Science and Manufacturing*, vol. 37, no. 5, pp. 778–789, 2006.

165. N. Valoroso and L. Champeny, "A damage-mechanics-based approach for modelling decohesion in adhesively bonded assemblies," *Eng. Fract. Mech.*, vol. 73, pp. 2774–2801, 2006.

166. L. Allix and A Corigliano, "Geometrical and interfacial non-linearities in the analysis of delamination in composites," *Int. J. Solids Struct.*, vol. 36, pp. 2189–2216, 1999

REFERENCES FOR THE PLATE AND SHELL INTERFACES

167. D. Chapelle and K.J. Bathe, *The Finite Element Analysis of Shells — Fundamentals*, Springer-Verlag, Berlin Heidelberg, 2003.

168. J.N. Reddy, *Mechanics of Laminated Composite Plates and Shells — Theory and Analysis*, CRC Press, Second Edition, 2004.

REFERENCES FOR THE MEMBRANE INTERFACE

169. D. Roddeman, J. Drukker, C. Oomens, and J. Janssen, "The Wrinkling of Thin Membranes: Part 1: Theory," *J. Appl. Mech.: Transactions of the ASME*, vol 54, pp. 884–887, 1987.

170. H. Schoop, L.Taenzer, and J. Hornig, "Wrinkling of Nonlinear Membranes," *Comp. Mech.*, vol. 29, pp. 68–74, 2002.

171. A. Patil, "Inflation and Instabilities of Hyperelastic Membranes," PhD thesis, KTH, 2016.

REFERENCES FOR THE PHASE FIELD IN SOLIDS INTERFACE

172. C. Miehe, F. Welschinger, M. Hofacker, "Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations," *Int. J. Numer. Methods Eng.,* vol. 83, pp. 1273–1311, 2010.

173. C. Miehe, L. Schänzel, H. Ulmer, "Phase field modeling of fracture in multi-physics problems. Part I. Balance of crack surface and failure criteria for brittle crack propagation in thermo-elastic solids," *Comput. Methods Appl. Mech. Eng.*, vol. 294, pp. 449–485, 2015.

174. J.–Y. Wu, V.P. Nguyen, C.T. Nguyen, D. Sutula, S. Sinaie, S.P.A. Bordas, "Phase-field modeling of fracture," *Adv. Appl. Mech.*, vol. 53, pp. 1–183, 2020.

4

Solid Mechanics

This chapter describes the Solid Mechanics interface, which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Solid Mechanics Interface

The **Solid Mechanics (solid)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is intended for general structural analysis of 3D, 2D, 1D, or axisymmetric bodies. In 2D, 1D, and 1D axisymmetry, plane stress, plane strain, or generalized plane strain assumptions can be used. The Solid Mechanics interface is based on solving the equations of motion together with a constitutive model for a solid material. Results such as displacements, stresses, and strains are computed.

The functionality provided by the Solid Mechanics interface depends on the products you are using. The Acoustics Module, MEMS Module, and Structural Mechanics Module add several features, for example geometric nonlinearity and advanced boundary conditions such as contact, follower loads, and nonreflecting boundaries.

The default material is a Linear Elastic Material. With either the Nonlinear Structural Materials Module or the Geomechanics Module, the physics interface is extended with more materials, for example, material models for plasticity, hyperelasticity, creep, and concrete. You can also add your own material models using an External Stress–Strain Relation

	For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, the following default nodes are also added to the **Model Builder**: **Linear Elastic Material**, **Free** (a boundary condition where boundaries are free, with no loads or constraints), and **Initial Values**. For axisymmetric models, an **Axial Symmetry** node is also added.

Then, from the **Physics** toolbar, you can add other nodes that implement, for example, solid mechanics material models, boundary conditions, and loads. You can also right-click **Solid Mechanics** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to

different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `solid`.

From the **2D approximation** list, select **Plane strain**, **Plane stress**, or **Generalized plane strain**.

Plane strain is relevant when the 2D model can be considered as a cut through an object that is infinitely long in the out-of-plane direction or as a soft object confined between rigid walls. The strain in the out-of-plane direction is assumed to be zero.

Plane stress is relevant for structures that are thin in the out-of-plane direction, such as a thin plate. When using plane stress, the Solid Mechanics interface solves for the out-of-plane strain displacement derivative, $\frac{\partial w}{\partial Z}$, in addition to the displacement field $\mathbf{u}$, in order to fulfill the condition that there is no stress in the thickness direction.

The generalized plane strain condition is similar to plane strain, but allows for a nonzero out-of-plane strain. It is representative for the central parts of a long object, which is stress free at the ends. In this case, you can choose between two assumptions: a uniform out-of-plane strain, or a full linear distribution of the out-of-plane strain. The latter assumption corresponds to bending in the out-of-plane direction, and is used when the **Enable out-of-plane bending** checkbox is selected.

For more information see the theory section.

When combining Solid Mechanics with other types of physics, there is often an assumption that the out-of-plane extension is infinitely long. This is the case in, for example, Acoustic-Structure interaction problems. In these cases, **Plane strain** is usually the correct choice.

Select **Out-of-plane mode extension (time-harmonic)** to prescribe an out-of-plane wave number to be used in mode analysis, eigenfrequency, and frequency domain studies. When selected, enter the **Out-of-plane wave number** k_z. The input value will only be taken into account in eigenfrequency and frequency domain studies. For mode analysis, the out-of-plane wave number is computed as an eigenvalue.

For more information, see Out-of-Plane Waves in the *Structural Mechanics Theory* chapter.

AXIAL SYMMETRY APPROXIMATION

Select **Include circumferential displacement** to add a dependent variable to account for the out-of-plane displacement. This can be used to model axial deformation with rotation around the axis of symmetry.

Select **Circumferential mode extension** to prescribe a circumferential wave number to be used in eigenfrequency or frequency domain studies. When selected, enter the **Azimuthal mode number** m.

For more information, see Circumferential Displacement and Out-of-Plane Waves in the *Structural Mechanics Theory* chapter.

This section is only available with certain COMSOL products (see www.comsol.com/products/specifications/).

Eigenfrequency Analysis of a Free Cylinder: Application Library path **Structural_Mechanics_Module/Verification_Examples/free_cylinder**.

From the **y direction** and **z direction** lists, select **Plane strain**, **Plane stress**, or **Generalized plane strain**.

Plane strain is relevant when the model can be considered as a cut through an object that is infinitely long in the corresponding out-of-plane direction (y or z), or as a soft object confined between rigid walls. The strain in the out-of-plane direction is assumed to be zero.

Plane stress is relevant for structures that are thin in the corresponding out-of-plane direction, such as a thin plate. When using plane stress, the Solid Mechanics interface solves for the out-of-plane strain displacement derivatives, $\frac{\partial v}{\partial Y}$ or $\frac{\partial w}{\partial Z}$, in addition to the displacement field $\mathbf{u}$, in order to fulfill the condition that there is no stress in the thickness direction.

The generalized plane strain condition is similar to plane strain, but allows for a constant nonzero out-of-plane strain. It is representative for the central parts of a long object, which is stress free at the ends.

For more information see the theory section.

When combining Solid Mechanics with other types of physics, there is often an assumption that the out-of-plane extension is infinitely long. This is the case in, for example, Acoustic-Structure interaction problems. In these cases, **Plane strain** is usually the correct choice.

From the **z direction** list, select **Plane strain**, **Plane stress**, or **Generalized plane strain**.

Plane strain is relevant when the model can be considered as a cut through a cylinder, which is infinitely long in the z direction, or as a soft object confined between rigid walls. The strain in the z direction is assumed to be zero.

Plane stress is relevant for thin discs. When using plane stress, the Solid Mechanics interface solves for the displacement derivative in the z direction, $\frac{\partial w}{\partial Z}$, in addition to the displacement field **u**, in order to fulfill the condition that there is no stress in the thickness direction.

The generalized plane strain condition is similar to plane strain, but allows for a nonzero strain in z direction. It is representative for the central parts of a long cylinder, which is stress free at the ends.

For more information see the theory section.

When combining Solid Mechanics with other types of physics, there is often an assumption that the out-of-plane extension is infinitely long. This is the case in, for example, Acoustic-Structure interaction problems. In these cases, **Plane strain** is usually the correct choice.

THICKNESS

For 2D and 1D axisymmetric components, enter a value or an expression for the thickness d. The default value of 1 m is suitable for plane strain models, where it represents a unit-thickness slice, for example. For plane stress models, consider entering the actual thickness.

When manually combining Solid Mechanics with other physics interfaces, you must make sure that the same thickness assumption is used everywhere. In most cases, the default settings will be correct since interfaces that do not have an explicit thickness property will implicitly assume unit thickness. When using other unit systems than the default SI system, you need to pay special attention to the values of this property.

Use a Change Thickness node to change thickness in parts of the geometry if necessary.

CROSS-SECTIONAL AREA

For 1D components, enter a value or an expression for the cross-section area A_c. The default value is $1\ \mathrm{m}^2$, which is suitable for models where plane strain assumptions are used in both transverse directions. For plane stress models, consider entering actual cross-section data.

When combining Solid Mechanics with other physics interfaces, you must make sure that the same cross-section assumption is used everywhere. In most cases, the default settings will be correct. When using other unit systems than the default SI system, you need to pay special attention to the values of this property.

Use a Change Cross Section node to change thickness in parts of the geometry if necessary.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasistatic (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural

frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

For problems with creep, and sometimes viscoelasticity, the problem can be considered as quasistatic. This is also the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

ENERGY DISSIPATION

From the **Store dissipation** list, select **Individual contributions**, **Total**, or **Off**.

Select **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Select **Total** to accumulate all the dissipative processes into one variable.

TRANSIENT SOLVER SETTINGS

In this section, you can add instructions used when generating the solver sequence for a wave-propagation transient problem. Select the **Maximum frequency to resolve** checkbox to activate this functionality.

Enter the **Maximum frequency to resolve** in the model, $f_{\text{max,sol}}$.

Select the **Time stepping** (method) as **Fixed (preferred)** or **Free**. The **Free** option is in general not recommended for wave problems.

The generated solver will be adequate in most situations if the computational mesh also resolves the frequency content in the model. Note that any changes made to these settings (after the model is solved the first time) will only be reflected in the solver if **Show Default Solver** or **Reset Solver to Defaults** is selected in the study.

For highly nonlinear problems with user-defined terms, manual tuning of the solver may be necessary. In nonlinear models, the maximum frequency to resolve should be selected based on the number of harmonics to be resolved.

TYPICAL WAVE SPEED FOR PERFECTLY MATCHED LAYERS

The typical wave speed c_{ref} is a parameter for the perfectly matched layers (PMLs) if used in a solid wave propagation model. The default value is `solid.cp`, the pressure-wave speed. To use another wave speed, enter a value or an expression in the **Typical wave speed for perfectly matched layers** field.

This section is only available with COMSOL products that include PMLs (see www.comsol.com/products/specifications/).

PORT SWEEP SETTINGS

Select to enable the **Activate port sweep** option. This option is used to compute the full scattering matrix when Port conditions are used. For more details see The Port Sweep Functionality subsection. The section only exists for 3D geometries.

AUTOMATED MODEL SETUP

This section will only be displayed if a mesh on NASTRAN® format, containing RBE2 elements, has been imported in an **Import** node under **Mesh**. The purpose is to automatically create rigid connectors from RBE2 elements in the NASTRAN file.

An RBE2 element represents a rigid connection between a set of mesh vertices. This means that it can, and often does, connect elements from different physics interfaces.

In the menu in the section title, you can select **Create Rigid Connectors from RBE2**. The effect is that one rigid connector will be created for each RBE2 element in the imported file. This will happen for all physics interfaces in the **Interfaces** list. Supported interfaces are: Solid Mechanics, Shell, Beam, and Multibody Dynamics. If there are RBE2 elements spanning more than one physics interface, they will be automatically connected.

The created rigid connectors will have point, edge, and boundary selections as inferred from the nodes in the RBE2 element and the mesh connectivity. The 'independent node' of the RBE2 element is used as center of rotation for the rigid connector.

The **Automated Model Setup** section is present in the Solid Mechanics, Shell, and Beam interfaces. In a model that contains several physics interfaces, you should use the automated model setup from only one of them, and make sure that all the involved interfaces are selected in the **Interfaces** list.

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. Normally these settings do not need to be changed.

You can choose how extra ODE variables added by some features are grouped in the **Dependent Variables** node of a generated solver sequence.

- Select the **Rigid materials** checkbox to group variables added by Rigid Material nodes.

- Select the **Rigid connectors** checkbox to group variables added by Rigid Connector nodes.
- Select the **Attachments** checkbox to group variables added by Attachment nodes.

The selection made in the **Advanced Settings** section can be overridden by the settings in the **Advanced** section of the Rigid Material, Rigid Connector or Attachment features.

DISCRETIZATION

In the Solid Mechanics interface, you can choose not only the order of the discretization, but also the type of shape functions: *Lagrange* or *serendipity*. For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will however give significant reductions of the model size for a given mesh containing hexahedral, prism, or quadrilateral elements. In 1D components, there is no difference between Lagrange and serendipity shape functions.

The default is to use **Quadratic serendipity** shape functions for the **Displacement field**. Using **Linear** shape functions will give what is sometimes called *constant stress* elements. Such a formulation will for many problems make the model overly stiff, and many elements may be needed for an accurate resolution of the stresses.

To display other settings for this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

!	See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

DEPENDENT VARIABLES

The physics interface uses the global spatial components of the **Displacement field** u as dependent variables. The default names for the components are (u, v, w) in 3D. In 2D the component names are (u, v), and in 2D axisymmetry they are (u, w). In 1D and 1D axisymmetry the default component name is (u). You can however not use the 'missing' component names in 2D or 1D as a parameter or variable name, since they are used internally.

You can change both the field name and the individual component names. If a new field name coincides with the name of another displacement field, the two fields (and the interfaces that define them) share degrees of freedom and dependent variable

component names. You can use this behavior to connect a Solid Mechanics interface to a Shell directly attached to the boundaries of the solid domain or to another Solid Mechanics interface sharing a common boundary.

A new field name must not coincide with the name of a field of another type (that is, it must contain a displacement field), or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

When a Solid Mechanics interface is active in the model, a set of default and predefined result features is created.

- A default plot named **Stress (solid)** containing the von Mises equivalent stress is created for all study types except Eigenfrequency, Buckling, or Boundary Mode Analysis.
- For the study types Eigenfrequency, Buckling, and Boundary Mode Analysis, a default plot named **Mode Shape (solid)** is created. The normalized modal displacement is shown.
- In the case of 2D axisymmetry, additional default plots are created using a revolved 3D representation. These plots are named **Stress (solid), 3D** and **Mode Shape (solid), 3D** respectively.
- For an Eigenfrequency study, a default evaluation group named **Eigenfrequencies** is created. In it, the computed eigenfrequencies are listed together with corresponding modal damping values.
- If a **Participation Factors** or **Response Spectrum** node is present under **Definitions** in the component, one additional default evaluation group named **Participation Factors** is created for an Eigenfrequency study.
- A result template named **Displacement (solid)** containing the norm of the displacement vector is created for all study types except Eigenfrequency, Buckling, or Boundary Mode Analysis.
- A result template named **Velocity (solid)** containing the norm of the velocity vector is created when either a time-domain or frequency-domain type of study is used.
- A result template named **Maximum Principal Tensile Stress (solid)** containing the largest principal stress is created for all study types except Eigenfrequency, Buckling, or Boundary Mode Analysis.
- A result template named **Maximum Principal Compressive Stress (solid)** containing the smallest principal stress is created for all study types except Eigenfrequency, Buckling, or Boundary Mode Analysis.
- A result template named **Modal Stress (solid)** is generated when an Eigenfrequency study is used.

- A number of result templates containing vector plots of various types of loads are created. The number of load plots depends on the types of load features that are active in the Solid Mechanics interface. The naming of the load groups follows the pattern **Boundary Loads (solid)**, **Point Loads (solid)**, and so on. They are all placed in a group node named **Applied Loads (solid)**.
- For 1D and 1D axisymmetric geometries, the same principles apply, but there are some differences. The most prominent one is that line graphs are used, so that for example more than one stress component can be shown.

There are also several nodes in the Solid Mechanics interface that can trigger the generation of specialized default or predefined result features. This is documented under the respective nodes.

See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

- Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics
- Solid Mechanics Theory
- Selecting Discretization

- *Stresses in a Pulley*: Application Library path **COMSOL_Multiphysics/Structural_Mechanics/stresses_in_pulley**
- *Eigenvalue Analysis of a Turbocharger Rotor*: Application Library path **COMSOL_Multiphysics/Structural_Mechanics/turbocharger_rotor**

Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics

The Solid Mechanics Interface has these domain, boundary, edge, point, and pair nodes and subnodes (listed in alphabetical order), which are available from the **Physics**

ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Solid Mechanics interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Domain** level are **Material Models**; **Volume Forces**; **Mass, Spring, and Damper**; **Domain Constraints**; and **More**.

The submenus at the **Boundary** level are **Connections**; **Mass, Spring, and Damper**; **More Constraints**; **Pairs**; and **More**.

There are also the **Edges** and **Points** submenus.

Note: Some submenus are only present with certain COMSOL products.

- Added Mass
- Antisymmetry
- Attachment
- Base Excitation
- Body Load
- Bolt Pretension
- Bolt Thread Contact
- Boundary Load
- Cell Periodicity
- Change Thickness
- Change Cross Section
- Contact
- Continuity
- Crack
- Edge Load
- Elastic Predeformation
- Elastoplastic Soil Material
- Enclosed Cavity
- External Stress–Strain Relation
- Fasteners
- Fixed Constraint
- Fixed Joint
- Free
- General Contact
- Gravity
- Hyperelastic Material
- Inertia Relief
- Initial Values
- Linear Elastic Material
- Linearly Accelerated Frame
- Local System Results
- Low-Reflecting Boundary
- Nonlinear Elastic Material
- Piezoelectric Material
- Periodic Condition
- Piezomagnetic Material
- Point Load
- Point Load, Free
- Point Load (on Axis)
- Point Load
- Point Mass
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Reduced Flexible Components
- Rigid Motion Suppression
- Rigid Connector
- Rigid Material
- Ring Load
- Ring Load, Free
- Roller
- Rotating Frame
- Section Forces
- Shape Memory Alloy
- Spring Foundation
- Spring–Damper
- Stress Linearization
- Symmetry
- Test Material

- Thin Layer
- Wave Speeds

	See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Below is a list of subnodes available from main parent nodes as indicated in the documentation:

- Activation
- Adhesion
- Applied Force (Rigid Connector)
- Applied Force (Rigid Material)
- Applied Moment (Rigid Connector)
- Applied Moment (Rigid Material)
- Bolt Selection
- Boundary Load (Thin Layer)
- Boundary Pair
- Center of Mass Nodes (Boundary, Edge, Point)
- Center of Rotation Nodes (Boundary, Edge, Point)
- Component Definition
- Conduction Loss (Time-Harmonic)
- Concrete
- Contact Model
- Crack Closure
- Creep
- Damage
- Damping
- Damping (General Contact)
- Decohesion
- Destination Filter
- Destination Point (for Spring–Damper)
- Dielectric Loss
- Elastic Predeformation
- External Strain
- External Stress
- Face Load (Crack)
- Face Load (Thin Layer)
- Fiber
- Fiber (Thin Layer)
- Filter (Enclosed Cavity)
- Fixed Constraint (Rigid Material)
- Fixed Constraint (Thin Layer)
- Fluid (Enclosed Cavity)
- Friction
- Hole Selection
- Hygroscopic Swelling
- Initial Stress and Strain
- Initial Values (Rigid Material)
- Intercalation Strain
- J-Integral
- Location Nodes (Boundary, Edge, Point)
- Mass and Moment of Inertia (Rigid Connector)
- Mass and Moment of Inertia (Rigid Material)
- Mechanical Damping
- Offset
- Phase
- Plasticity
- Porous Plasticity
- Predeformation
- Prescribed Displacement (Thin Layer)
- Prescribed Displacement/Rotation
- Prescribed Pressure (Enclosed Cavity)
- Pressure-Dependent Plasticity
- Reverse Crack Front
- Rocks
- Roller (Thin Layer)
- Safety
- Safety (for Fasteners)
- Set Variables

- Slip Velocity
- Soil Plasticity
- Source Filter
- Source Point (for Spring–Damper)
- Spring Foundation (Rigid Connector)
- Spring Foundation (Rigid Material)
- Spring Material
- Stabilization
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Attachment)
- Thermal Expansion (Fiber)
- Thermal Expansion (Rigid Connector)
- Virtual Crack Extension
- Viscoelasticity
- Viscoplasticity
- Wear

Initial Values

The **Initial Values** node adds initial values for the displacement field and structural velocity field that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. In addition to the default **Initial Values** node always present in the interface, you can add more **Initial Values** nodes if needed.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Initial Values node.

INITIAL VALUES

Enter values or expressions for the initial values of the **Displacement field u** (the displacement components u, v, and w in 3D), and the **Structural velocity field** $\partial\mathbf{u}/\partial t$.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Initial Values**

Solid Mechanics, Explicit Dynamics > **More** > **Initial Values**

Layered Shell > **More** > **Initial Values**
Membrane > **More** > **Initial Values**
Truss > **More** > **Initial Values**
Truss, Explicit Dynamics > **More** > **Initial Values**
Wire > **More** > **Initial Values**

Ribbon
Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Domains > **More** > **Initial Values**

Physics tab with **Layered Shell** or **Membrane** selected:

Boundaries > **More** > **Initial Values**

Physics tab with **Truss**; **Truss, Explicit Dynamics;** or **Wire** selected:

Edges > **More** > **Initial Values**

Change Thickness

Use the **Change Thickness** node to model domains with a thickness other than the overall thickness defined in the physics interface's Thickness section. The **Change Thickness** node is only available in 2D, and 1D axisymmetry.

CHANGE THICKNESS

Enter a value for the **Thickness** d. This value replaces the overall thickness for the selected domains.

LOCATION IN USER INTERFACE

Context Menus
Solid Mechanics > **More** > **Change Thickness**
Multibody Dynamics > **More** > **Change Thickness**

Ribbon
Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Domains > **More** > **Change Thickness**

Change Cross Section

Use the **Change Cross Section** node to model domains with a cross section other than the overall cross section defined in the physics interface's Cross-Sectional Area section. The **Change Cross Section** node is only available in 1D.

CHANGE CROSS SECTION

Enter a value for the cross section area A_c. This value replaces the overall cross section area for the selected domains.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Change Cross Section**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **More** > **Change Cross Section**

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic solid and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Damping
- Viscoelasticity
- Plasticity
- Pressure-Dependent Plasticity
- Creep
- Viscoplasticity
- Porous Plasticity
- Soil Plasticity
- Concrete
- Rocks
- Fiber
- Damage
- Activation
- Safety

Note: Some options are only available with certain COMSOL products (see www.comsol.com/products/specifications/). Also, the available options depend on the physics interface in which the **Linear Elastic Material** is used.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Linear Elastic Material node. The way the **Linear Elastic Material** node interacts with material definitions differ significantly between the Layered Shell interface and the other physics interfaces.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

LINEAR ELASTIC MATERIAL

Define the **Material symmetry** and the linear elastic material properties.

Material Symmetry

Select the **Material symmetry** — **Isotropic**, **Orthotropic**, **Anisotropic**, or **Crystal**. Select:

- **Isotropic** for a material that has the same properties in all directions.
- **Orthotropic** for a material that has different material properties in orthogonal directions. It is also possible to define **Transversely isotropic** material properties.
- **Anisotropic** for a material that has different material properties in different directions.
- **Crystal** for a material that has certain crystal symmetry.

The **Crystal** symmetry is only available in the Solid Mechanics interface.

In the Solid Mechanics, Explicit Dynamics interface, the only available option is **Isotropic**.

Note: The **Orthotropic**, **Anisotropic**, and **Crystal** options are only available with certain COMSOL products (see www.comsol.com/products/specifications/)

	In the Layered Shell interface, the chosen material symmetry applies to all selected layers, irrespective of whether the material data is entered explicitly as **User defined** in the **Linear Elastic Material** node, or is obtained from a **Layered Material** node using the default **From material** option.

	• Material Models • Linear Elastic Material • Orthotropic and Anisotropic Materials • Crystal Symmetry

Density

The default **Density** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

	The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

	See also • Mass Density and Volume Reference Temperature. • Using Common Model Input • Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

Specification of Elastic Properties for Isotropic Materials

For an **Isotropic** material, from the **Specify** list select a pair of elastic properties for an isotropic material — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and shear-wave speeds**. For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another according to Table 4-1.

TABLE 4-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Young's modulus	E =	E	E	$\frac{9KG}{3K+G}$	$\mu\frac{3\lambda+2\mu}{\lambda+\mu}$
Poisson's ratio	ν =	ν	$\frac{E}{2G}-1$	$\frac{1}{2}\left(1-\frac{3G}{3K+G}\right)$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	K =	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	K	$\lambda+\frac{2\mu}{3}$
Shear modulus	G =	$\frac{E}{2(1+\nu)}$	G	G	μ
Lamé parameter λ	λ =	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K-\frac{2G}{3}$	λ
Lamé parameter μ	μ =	$\frac{E}{2(1+\nu)}$	G	G	μ
Pressure-wave speed	c_p =	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\frac{K+4G/3}{\rho}}$	$\sqrt{\frac{\lambda+2\mu}{\rho}}$
Shear-wave speed	c_s =	$\sqrt{\frac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

The individual property parameters are:

- **Young's modulus** (elastic modulus) E.
- **Poisson's ratio** ν.
- **Shear modulus** G.
- **Bulk modulus** K.
- **Lamé parameter** λ and **Lamé parameter** μ.

- **Pressure-wave speed** (longitudinal wave speed) c_p.
- **Shear-wave speed** (transverse wave speed) c_s. This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Specification of Elastic Properties for Orthotropic Materials

When **Orthotropic** is selected from the **Material symmetry** list, the material properties are different in orthogonal directions (principal directions) given by the axes of the selected coordinate system. The **Material data ordering** can be specified in either **Standard** or **Voigt** notation. When **User defined** is selected, enter three values in the fields for **Young's modulus** $\mathbf{E}$, **Poisson's ratio** ν, and the **Shear modulus** $\mathbf{G}$. The latter defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$

The Poisson's ratio ν_{ij} are defined differently depending on the application field. It is easy to transform among definitions, check which one the reference material uses.

You can set an orthotropic material to be **Transversely isotropic**. Then, one principal direction in the material is different from two others that are equivalent. This special direction is assumed to be the first axis of the selected coordinate system. Because of the symmetry, the following relations hold:

$$E_3 = E_2$$

$$\nu_{13} = \nu_{12}$$

$$G_{23} = \frac{E_2}{2(1+\nu_{23})}$$

$$G_{12} = G_{13}$$

Thus, only five elasticity moduli need to be entered when the **User defined** option is selected.

Specification of Elastic Properties for Anisotropic Materials

When **Anisotropic** is selected from the **Material symmetry** list, the material properties vary in all directions. They can be specified using either the **Elasticity matrix**, $\mathbf{D}$ or the **Compliance matrix**, $\mathbf{D}^{-1}$. Both matrices are symmetric. The **Material data ordering** can be specified in either **Standard** or **Voigt** notation. When **User defined** is selected, a 6-by-6 symmetric matrix is displayed.

In 1D and 1D axisymmetry, the elasticity matrix is assumed to represent either isotropic or orthotropic material. Entering components in the elasticity matrix that couple extension and shear, for instance, should be avoided.

Specification of Elastic Properties for Crystals

Because of the material symmetry, only certain components of the elasticity matrix need to be specified. The actual components to enter depend on the selected **Crystal system** — **Cubic (3 constants)**, **Hexagonal (5 constants)**, **Trigonal (6 constants)**, **Trigonal (7 constants)**, **Tetragonal (6 constants)**, **Tetragonal (7 constants)**, or **Orthorhombic (9 constants)**.

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which adds an extra dependent variable for either the pressure or for the volumetric strain. For details, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

The mixed formulations are not available in the Solid Mechanics, Explicit Dynamics interface.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, or **Strain formulation**. It is also possible to select an **Implicit formulation** when an assumption of plane stress is used.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

This section is only available with COMSOL products that support geometrically nonlinear analysis (see www.comsol.com/products/specifications/).

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

DISCRETIZATION

If **Pressure formulation** is used, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**. If **Strain formulation** is used, select the discretization for the **Auxiliary volumetric strain** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

The **Discretization** section is available when **Pressure formulation** or **Strain formulation** is selected from the **Use mixed formulation** list. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the

stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

ESTIMATE OF MAXIMUM WAVE SPEED

This section is only present in the Solid Mechanics, Explicit Dynamics interface.

Select how the maximum wave speed in the material is determined — **Automatic** or **User defined**. For **User defined**, enter the maximum wave speed, c_{max}.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Linear Elastic Material**
Solid Mechanics, Explicit Dynamics > **Material Models** > **Linear Elastic Material**
Layered Shell > **Material Models** > **Linear Elastic Material**
Multibody Dynamics > **Material Models** > **Linear Elastic Material**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Material Models** > **Linear Elastic Material**

Physics tab with **Layered Shell** selected:

Boundaries > **Material Models** > **Linear Elastic Material**

Physics tab with **Multibody Dynamics** selected:

Domains > **Material Models** > **Linear Elastic Material**

Nonlinear Elastic Material

The **Nonlinear Elastic Material** feature is used to model stress–strain relationships which are nonlinear even at infinitesimal strains. It is available in the Solid Mechanics and Membrane interfaces. This material model requires either the Nonlinear Structural Materials Module or the Geomechanics Module. **Nonlinear Elastic Material** is available for 3D, 2D, and 2D axisymmetry.

By adding the following subnodes to the **Nonlinear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Porous Plasticity
- Soil Plasticity
- Concrete
- Rocks
- Fiber
- Safety

Note: Some options are only available with certain COMSOL products (see www.comsol.com/products/specifications/)

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

NONLINEAR ELASTIC MATERIAL

The available material models depend on the COMSOL products you are using.

Nonlinear Structural Materials Module: Select a **Material model**: **Ramberg–Osgood**, **Power law**, **Uniaxial data**, **Shear data**, **Bilinear elastic**, or **User defined**.

Geomechanics Module: Select a **Material model**: **Ramberg–Osgood**, **Hyperbolic law**, **Hardin–Drnevich**, **Duncan–Chang**, **Duncan–Selig**, **Small-strain overlay**, or **User defined**.

Density

All nonlinear elastic material models have density as an input. The default **Density** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

See also

- Mass Density and Volume Reference Temperature
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which adds an extra dependent variable for either the pressure or for the volumetric strain, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, or **Strain formulation**. It is also possible to select an **Implicit formulation** when an assumption of plane stress is used.

Ramberg–Osgood, Power Law, Hyperbolic Law, Hardin–Drnevich, Duncan–Chang, Duncan–Selig, or Small-Strain Overlay

Select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

From the **Specify** list select a pair of elastic properties for an isotropic material — **Young's modulus and Poisson's ratio** (the default for Ramberg–Osgood, Power law, Duncan–Chang, and Duncan–Selig) or **Bulk modulus and shear modulus** (the default for Hyperbolic law and Hardin–Drnevich).

Then depending on the selections, define the applicable parameters:

- **Young's modulus** E.
- **Poisson's ratio** ν.
- **Shear modulus** G.
- **Bulk modulus** K.
- For **Ramberg–Osgood**:
  - **Reference stress** σ_{ref}.
  - **Reference strain** ε_{ref}.
  - **Stress exponent** n.
- For **Power law** and **Hyperbolic law**:
  - **Reference shear strain** γ_{ref}.
  - **Strain exponent** n.
- For **Hardin–Drnevich**, define the **Reference shear strain** γ_{ref}.
- For **Duncan–Chang**, define the **Ultimate deviatoric stress** q_{ult}.
- For **Duncan–Selig**:
  - **Ultimate deviatoric stress** q_{ult}.
  - **Ultimate strain** ε_{ult}.
- For **Small-strain overlay**:
  - **Initial shear modulus** G_0
  - **Reference shear strain** γ_{ref}

Uniaxial Data

For **Uniaxial data** the **Uniaxial stress function** σ_{ax} uses the value **From material** (if it exists) or **User defined**. If **User defined** is selected from the list, the default expression for σ_{ax} is the linear function `210[GPa]*<physics>.eax`, which corresponds to a linear elastic

material with a Young's modulus of 210 GPa. The variable `<physics>.eax` corresponds to the elastic uniaxial strain in pure axial loading, and is named using the scheme `<physics>.eax`, for example, `solid.eax`.

From the **Specify** list select how to specify the second elastic property for the material — **Bulk modulus** or **Poisson's ratio**. Then, depending on the selection, enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression:

- **Bulk modulus** K.
- **Poisson's ratio** ν.

When you select **Bulk modulus**, the Young's modulus is computed from the tensile part of the **Uniaxial stress function** σ_{ax}. When you select **Poisson's ratio**, you can either use the tensile part (default), or use the full tensile-compressive curve by selecting the checkbox **Use nonsymmetric stress–strain data**.

Shear Data

For **Shear data** the **Shear stress function** τ uses the value **From material** (if it exists) or **User defined**. If **User defined** is selected from the list, the default expression for τ is the linear function `80[GPa]*<physics>.esh`, which corresponds to a linear elastic material with a shear modulus of 80 GPa. The variable `<physics>.esh` corresponds to the elastic shear strain in pure shear loading, and it is named using the scheme `<physics>.esh`, for example, `solid.esh`.

The default **Bulk modulus** K uses values **From material**. For **User defined** enter another value or expression.

Bilinear Elastic

For **Bilinear elastic** enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

- **Bulk modulus in tension** K_t.
- **Bulk modulus in compression** K_c.
- **Shear modulus** G.

Small-Strain Overlay

In case of cyclic loading the load reversal points are automatically detected, but in many cases these are known a priori. The **Load Reversal Points** section enables to set the load reversal points manually based on user defined expressions.

From the list select how to specify the load reversal points — **Automatic**, **None**, or **User defined**. For **User defined** enter a Boolean expression to fulfill the reversal point for each component of the strain tensor.

User Defined

In the **User defined** material model, you specify the bulk modulus implicitly by entering the relation between pressure and volumetric elastic strain. Enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

- **Pressure** p. The default expression is `(-160[GPa])*solid.eelvol`, which corresponds to a linear elastic material with a bulk modulus of 160 GPa.
- **Shear modulus** G.

See also Nonlinear Elastic Materials in the *Structural Mechanics Theory* chapter.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

Only the additive decomposition of strains is available for the Small-Strain Overlay model.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

DISCRETIZATION

If **Pressure formulation** is used, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**. If **Strain formulation** is used, select the discretization for the **Auxiliary volumetric strain** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

The **Discretization** section is available when **Pressure formulation** or **Strain formulation** is selected from the **Use mixed formulation** list. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Nonlinear Elastic Material**
Membrane > **Material Models** > **Nonlinear Elastic Material**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Nonlinear Elastic Material**

Physics tab with **Membrane** selected:

Boundaries > **Material Models** > **Nonlinear Elastic Material**

Elastoplastic Soil Material

The **Elastoplastic Soil Material** feature is used to model stress–strain relationships that are nonlinear even at infinitesimal strains. It is available in the Solid Mechanics interface. This material model requires the Geomechanics Module (see www.comsol.com/products/specifications/). **Elastoplastic Soil Material** is available for 3D, 2D, and 2D axisymmetry.

By adding the following subnodes to the **Elastoplastic Soil Material** node you can incorporate other effects:

- Thermal Expansion (for Materials)
- Initial Stress and Strain
- External Stress
- Damping
- Set Variables
- Safety
- Cap and Cutoff

Add an **External Stress** node in case you need to define a pore pressure in a porous soil. The **Pore pressure** p_A is user defined by default. The default value is 1 atm, but you can change it to another value or expression for the pore fluid pressure. If there are other

physics interfaces (like Darcy's Law) in the model that make a pressure variable available, such variables will be available in the list.

When a **Porous Plasticity** or **Elastoplastic Soil Material** node is present, a **Volumetric Plastic Strain** plot and a **Void Ratio** plot are available under **Result Templates**.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

ELASTOPLASTIC SOIL MATERIAL

Select a **Material model** from the list: **Modified Cam-clay**, **Modified structured Cam-clay**, **Extended Barcelona basic**, **Hardening soil**, or **Hardening soil small strain**.

Density

All elastoplastic soil models have density as an input. The default **Density** ρ uses values **From material**. For **User defined** enter a custom value or expression.

If the material has a temperature dependent density or other material property, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties, see Computing Mass Properties.

See also

- Mass Density and Volume Reference Temperature
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*.

Octahedral Section

Select the **Octahedral section** — **Circular**, **Mohr–Coulomb**, or **Matsuoka–Nakai**. The shape of the yield function in the Octahedral Plane is generally circular, but it can be made a function of the Lode angle θ. See Octahedral Section for details.

When the octahedral section is **Mohr–Coulomb**, the slope of the critical state line, M, is computed from the friction angle ϕ.

When the octahedral section is **Circular** or **Matsuoka–Nakai**, select how the slope of the critical state line, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion and Matsuoka–Nakai Criterion for details.

Plastic Potential

Select the **Plastic potential** Q_{p} related to the flow rule — **Associated**, **Nonassociated**, or **McDowell–Hau**.

When **Nonassociated** is selected in the **Plastic potential** list, the dilatation angle ψ replaces the friction angle ϕ in the plastic potential Q_{p}. The default value for the **Dilatation angle** ψ is taken **From material**.

When **McDowell–Hau** is selected in the **Plastic potential** list, the default for the **Plastic potential shape parameter** ζ is taken **From material**.

Equivalent Plastic Strain

Select how the **Equivalent plastic strain** $\varepsilon_{\mathrm{pe}}$ is computed — **Associated**, **von Mises**, or **User defined**. Enter a **User defined** value in the h_{p} field as needed. See Hardening Rule for details.

Modified Cam-Clay

From the **Specify** list select how to specify the elastic property for the material — **Poisson's ratio** or **Shear modulus**. Then, depending on the selection, enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

The defaults for the **Swelling index** κ, **Compression index** λ, **Friction angle** ϕ, and **Initial void ratio** e_0 are taken **From material**. For **User defined** enter other values or expressions.

Select how the **Slope of critical state line**, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion for details.

When **Nonassociated** is selected in the **Plastic potential** list, select how the **Slope of critical state line for plastic potential**, M_Q, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. Enter the **Dilatation angle** ψ to define the nonassociated plastic potential Q_p.

Enter a value or an expression for the **Reference pressure** p_{ref} and the **Preconsolidation pressure** p_{c0}.

For the **Initial void ratio**, you can alternatively select **From void ratio at reference pressure**, which then computes the initial void ratio from the reference pressure, the void ratio at reference pressure, the preconsolidation pressure, and the swelling and compression indexes. Then select the **Void ratio at reference pressure** e_{ref} as **From material** or **User defined**.

See also The Modified Cam-Clay Soil Model in the *Structural Mechanics Theory* chapter.

Isotropic Compression with Modified Cam-Clay Material Model: Application Library path **Geomechanics_Module/Verification_Examples/isotropic_compression**

Modified Structured Cam-Clay

From the **Specify** list select how to specify the elastic property for the material — **Poisson's ratio** or **Shear modulus**. Then, depending on the selection, enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

The defaults for the **Swelling index for structured clay** κ_s, **Compression index for destructured clay** λ_d, **Friction angle** ϕ, **Initial structure strength** p_{b0}, **Destructuring index for volumetric deformation** d_v, **Destructuring index for shear deformation** d_s, **Plastic deviatoric strain at failure** e_f^p, **Initial void ratio for structured clay** e_0, and **Additional void ratio at preconsolidation pressure** Δe_{c0} are taken **From material**. For **User defined** enter other values or expressions.

Select how the **Slope of critical state line**, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion for details.

When **Nonassociated** is selected in the **Plastic potential** list, select how the **Slope of critical state line for plastic potential**, M_Q, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. Enter the **Dilatation angle** ψ to define the nonassociated plastic potential Q_p.

Enter a value or an expression for the **Reference pressure** p_{ref} and the **Preconsolidation pressure** p_{c0}.

For the **Initial void ratio**, you can alternatively select **From void ratio at reference pressure for destructured clay**, which then computes the initial void ratio from the reference pressure, the void ratio at reference pressure for destructured clay, the preconsolidation pressure, and the swelling and compression indexes. Select the **Void ratio at reference pressure for destructured clay** $e_{ref,d}$ as **From material** or **User defined**.

See also The Modified Structured Cam-Clay Soil Model in the *Structural Mechanics Theory* chapter.

Extended Barcelona Basic

From the **Specify** list select how to specify the elastic property for the material — **Poisson's ratio** or **Shear modulus**. Then, depending on the selection, enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

The defaults for the **Swelling index at saturation** κ_0, **Swelling index for changes in suction** κ_s, **Compression index at saturation** λ_0, **Compression index for changes in suction** λ_s, **Friction angle** ϕ, **Weight parameter** w, **Soil stiffness parameter** m, **Plastic potential smoothing parameter** b, **Tension to suction ratio** k, **Initial yield value for suction** s_{y0}, and **Initial void ratio** e_0 are taken **From material**. For **User defined** enter other values or expressions.

Select how the **Slope of critical state line**, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion for details.

When **Nonassociated** is selected in the **Plastic potential** list, select how the **Slope of critical state line for plastic potential**, M_Q, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. Enter the **Dilatation angle** ψ to define the nonassociated plastic potential Q_p.

Enter a value or an expression for the **Initial suction** s_0, **Suction** s, **Reference pressure** p_{ref}, and **Preconsolidation pressure** p_{c0}.

For the **Initial void ratio**, you can alternatively select **From void ratio at reference pressure and saturation**, which then computes the initial void ratio from the reference pressure, the void ratio at reference pressure and saturation, the preconsolidation pressure, and the swelling and compression indexes. Then select the **Void ratio at reference pressure and saturation** $e_{\text{ref},0}$ as **From material** or **User defined**.

See also The Extended Barcelona Basic Soil Model in the *Structural Mechanics Theory* chapter.

Hardening Soil

Select the **Mobilized dilatancy angle** ψ_m — **Soreide**, **Rowe**, **Modified Rowe**, **Wehnert**, or **User defined**. See Mobilized Dilatancy Angle for details.

The **Reference initial stiffness for primary loading** E_i^{ref}, **Reference stiffness for unloading and reloading** E_{ur}^{ref}, **Poisson's ratio** ν, **Stress exponent** m, **Cohesion** c, **Friction angle** ϕ, and **Initial void ratio** e_0 are taken **From material**. For **User defined** enter other values or expressions.

Select how the **Slope of critical state line**, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion for details.

When **Nonassociated** is selected in the **Plastic potential** list, select how the **Slope of critical state line for plastic potential**, M_Q, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. Enter the **Dilatation angle** ψ to define the nonassociated plastic potential Q_p.

Enter a value or an expression for the **Failure ratio** R_f and the **Reference pressure** p_{ref}.

Activate the **Dilatancy cutoff** if needed. Enter a value or an expression for the **Maximum void ratio** $e_{\max}$ and the **Initial volumetric strain** ε_{vol0}.

If required, add a Cap and Cutoff subnode.

For the **Reference initial stiffness for primary loading** $E_{\mathrm{i}}^{\mathrm{ref}}$, you can alternatively select **From reference failure stiffness**, which then calculates $E_{\mathrm{i}}^{\mathrm{ref}}$ from the **Reference failure stiffness** $E_{50}{}^{\mathrm{ref}}$.

See also The Hardening Soil Model in the *Structural Mechanics Theory* chapter.

Hardening Soil Small Strain

Select the **Mobilized dilatancy angle** ψ_{m} — **Soreide**, **Rowe**, **Modified Rowe**, **Wehnert**, or **User defined**. See Mobilized Dilatancy Angle for details.

The defaults for the **Reference initial stiffness for primary loading** $E_{\mathrm{i}}^{\mathrm{ref}}$, **Reference stiffness for unloading and reloading** $E_{\mathrm{ur}}{}^{\mathrm{ref}}$, **Small strain shear modulus** G_0, **Poisson's ratio** ν, **Reference shear strain** γ_{ref}, **Stress exponent** m, **Cohesion** c, **Friction angle** ϕ, and **Initial void ratio** e_0 are taken **From material**. For **User defined** enter other values or expressions.

Select how the **Slope of critical state line**, M, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. See Mohr–Coulomb Criterion for details.

When **Nonassociated** is selected in the **Plastic potential** list, select how the **Slope of critical state line for plastic potential**, M_Q, is computed — **From material**, **Match to Mohr–Coulomb criterion**, or **User defined**. Enter the **Dilatation angle** ψ to define the nonassociated plastic potential Q_{p}.

Enter a value or an expression for the **Failure ratio** R_{f} and the **Reference pressure** p_{ref}.

Activate the **Dilatancy cutoff** if needed. Enter a value or an expression for the **Maximum void ratio** e_{max} and the **Initial volumetric strain** $\varepsilon_{\mathrm{vol0}}$.

If required, add a Cap and Cutoff subnode.

For the **Reference initial stiffness for primary loading** $E_{\mathrm{i}}^{\mathrm{ref}}$, you can alternatively select **From reference failure stiffness**, which then calculates $E_{\mathrm{i}}^{\mathrm{ref}}$ from the **Reference failure stiffness** $E_{50}{}^{\mathrm{ref}}$.

For the **Small strain shear modulus** G_0, you can alternatively select **From reference small strain shear modulus**, which then calculates G_0 from the **Reference small strain shear modulus** $G_0{}^{\mathrm{ref}}$.

See also The Hardening Soil Small Strain Model in the *Structural Mechanics Theory* chapter.

Load Reversal Points

In case of cyclic loading the load reversal points are automatically detected, but in many cases these are known a priori. The **Load Reversal Points** section enables to set the load reversal points manually based on user defined expressions.

From the list select how to specify the load reversal points — **Automatic**, **None**, or **User defined**. For **User defined** enter a Boolean expression to fulfill the reversal point for each component of the strain tensor.

NONLOCAL PLASTICITY MODEL

Nonlocal plasticity can be used to facilitate, for example, the modeling of material softening. Typical examples that involve material softening are finite-strain plasticity and soil plasticity. In these situations, standard (local) plasticity calculations reveal a mesh and topology dependence, where a mesh refinement fails to reproduce a physically sound solution. Nonlocal plasticity adds regularization to the equivalent plastic strain, thereby stabilizing the solution.

The default setting is **None**. Select **Implicit Gradient** to add nonlocal regularization to the equivalent plastic strain. Then, enter values or expressions for:

- **Length scale**, l_{int} — The regularization length scale for the equivalent plastic strain $\varepsilon_{\mathrm{pe}}$ should ideally be greater than the largest mesh element size.
- **Nonlocal coupling modulus**, H_{nl} — The value represents the penalization of the difference between the local and nonlocal variables. A larger value enforces the equivalent plastic strain $\varepsilon_{\mathrm{pe}}$ to be closer to the nonlocal equivalent plastic strain

$\varepsilon_{pe,nl}$. A typical value would be around 10% the equivalent shear modulus of the material.

See also Nonlocal Plasticity in the *Structural Mechanics Theory* chapter.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.

- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

Only the additive decomposition of strains is available for the Hardening Soil Small Strain model.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog, and it is available with the **Implicit gradient** nonlocal plasticity model.

Select the shape function for the **Nonlocal equivalent plastic strain** $\varepsilon_{pe,nl}$ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster

computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The plastic potential includes a vertex where it intersects the hydrostatic axis. To improve robustness of the plasticity algorithm, smoothing is applied to remove these singularities.

Select the **Smoothing of plastic potential** — **Automatic**, **Manual tuning**, or **User defined** to control how much the smoothed plastic potential deviates from the original potential.

When **Manual tuning** or **User defined** is selected, enter the **Edge smoothing parameter** β when using Mohr–Coulomb plastic potential. The default value is $\beta = 0.99$.

When **Manual tuning** is selected, enter a value for the **Vertex smoothing multiplier** f_{v}. This multiplier scales the amount of smoothing at the apex given by the **Automatic** option.

The **User defined** option allows full control of the smoothing by entering a value for the **Vertex smoothing parameter** $\sigma_{v,off}$. A typical value would be around 10% of the initial cohesion for soil models.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** or **Backward Euler, damped** is selected, it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

When the **Backward Euler, damped** method is selected, the Newton's method is enhanced by line search iterations. Using this method can improve the robustness of the plasticity algorithm when the plastic potential or hardening model are highly nonlinear. When selected, it is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

See also Apex Smoothing, Vertex Smoothing, and Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

Select the **Hardening function offset** — **Automatic** or **User defined** to control how much the smoothed potential at the apex deviates from the original plastic potential. The **Automatic** option does not apply any smoothing. Select **User defined** to enter a value. See Apex Smoothing for details.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Elastoplastic Soil Material**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Elastoplastic Soil Material**

Hyperelastic Material

The **Hyperelastic Material** subnode adds the equations for hyperelasticity at large strains. Hyperelastic materials can be suitable for modeling rubber and other polymers, biological tissue, and also for applications in acoustoelasticity. The **Hyperelastic Material** is available in the Solid Mechanics, Layered Shell, Shell, and Membrane interfaces. **Hyperelastic Material** is available for 3D, 2D, 2D axisymmetry, 1D, and 1D axisymmetry.

When a hyperelastic material is included in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** checkbox in the study settings is selected and cannot be cleared.

By adding the following subnodes to the **Hyperelastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Polymer Viscoplasticity
- Fiber
- Mullins Effect
- Damage

See also Hyperelastic Materials in the *Structural Mechanics Theory* chapter.

The **Hyperelastic Material** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present when **Hyperelastic Material** is used in the Layered Shell interface. See the documentation for the Hyperelastic Material node in the Layered Shell chapter.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

HYPERELASTIC MATERIAL

Select a hyperelastic **Material model** from the list and then go to the applicable section for more information.

- Neo-Hookean
- Saint-Venant–Kirchhoff
- Mooney–Rivlin, Two Parameters
- Mooney–Rivlin, Five Parameters
- Mooney–Rivlin, Nine Parameters
- Yeoh
- Ogden
- Storakers
- Varga
- Arruda–Boyce
- Gent
- van der Waals
- Blatz–Ko
- Gao
- Murnaghan
- Delfino
- Fung
- Extended Tube
- User Defined

In the Solid Mechanics, Explicit Dynamics interface:

- The material model **User defined** is not available.
- The options in the **Compressibility** list is, for all material models, limited to **Compressible, coupled** and **Compressible, uncoupled**.

Density

All hyperelastic material models have density as an input. The default **Density** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can

also select **User defined** to enter a value or an expression for the reference temperature locally.

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which adds an extra dependent variable for either the pressure or for the Lagrange multiplier to enforce incompressibility, see the Nearly Incompressible Hyperelastic Materials and Incompressible Hyperelastic Materials sections in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None** or **Pressure formulation**. It is also possible to select an **Implicit formulation** when an assumption of plane stress is used.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

Neo-Hookean

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, coupled**; **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, coupled** option is selected, specify the **Volumetric strain energy density** — **Simo–Pister** or **Miehe**.
- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.

- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\mathrm{el}} = 1$.

From the **Specify** list select a pair of elastic properties for the isotropic hyperelastic material — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and shear-wave speeds**. Each of these pairs define the Lamé parameters at infinitesimal deformation as it is possible to convert from one set of properties to another, see Specification of Elastic Properties for Isotropic Materials. For each property, select from the applicable list to either use the value **From material** or enter a **User defined** value or expression.

Saint-Venant–Kirchhoff

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, coupled**; **Compressible, uncoupled**; or **Nearly incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.

From the **Specify** list select a pair of elastic properties for the isotropic hyperelastic material — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and shear-wave speeds**. For each pair of properties, select from the applicable list to either use the value **From material** or enter a **User defined** value or expression. Each of these pairs define the Lamé parameters at infinitesimal deformation as it is possible to convert from one set of properties to another, see Specification of Elastic Properties for Isotropic Materials.

Mooney–Rivlin, Two Parameters

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

The **Model parameters** C_{10} and C_{01} use values **From material**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

Mooney–Rivlin, Five Parameters

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

The **Model parameters** C_{10}, C_{01}, C_{20}, C_{02}, and C_{11} all use values **From material**.

Mooney–Rivlin, Nine Parameters

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

The **Model parameters** C_{10}, C_{01}, C_{20}, C_{02}, C_{11}, C_{30}, C_{03}, C_{21}, and C_{12} all use values **From material**.

Yeoh

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

The **Model parameters** c_1, c_2, and c_3 all use values **From material**.

Ogden

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

In the table for the **Ogden parameters**, enter values or expressions in each column: **Shear modulus (Pa)** and **Alpha parameter**.

Storakers

For **Storakers**, in the table for the **Storakers parameters**, enter values or expressions in each column: **Shear modulus (Pa)**, **Alpha parameter**, and **Beta parameter**.

Varga

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

The **Model parameters** c_1, c_2, and c_3 all use values **From material**.

Arruda–Boyce

From the **Isochoric strain energy** list select how the strain energy density of the material is specified in terms of the invariants — **Five terms expansion** or **Inverse Langevin function**.

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\mathrm{el}} = 1$.

The default values for the **Macroscopic shear modulus** μ_0 and the **Number of segments** N use values **From material**.

Gent

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\mathrm{el}} = 1$.

The default values for the **Macroscopic shear modulus** μ and the model parameter j_{m} use values **From material**.

van der Waals

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{\text{el}} = 1$.

The default values for the **Shear modulus** μ, the **Maximum chain stretch** λ_{m}, the **Chain network interaction** α, and the **Weight** β use values **From material**.

Blatz–Ko

For **Blatz–Ko** the **Shear modulus** μ and the **Model parameters** β and ϕ all use values **From material**.

Gao

For **Gao** the **Model parameters** a and n use values **From material**.

Murnaghan

For **Murnaghan** the **Murnaghan third-order elastic moduli** constants l, m, and n and the **Lamé parameters** λ and μ use values **From material**.

Delfino

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**.

The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.

- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{el} = 1$.

The **Model parameters** a and b use values **From material**.

Fung

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, coupled**; **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.
- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{el} = 1$.

The **Coefficient matrix** $\mathbf{A}$ and **Fung parameter** c use values **From material**.

The **Coefficient matrix** $\mathbf{A}$ provides the anisotropic material properties in the directions given by the **Coordinate system** list. The **Material data ordering** can be specified in either **Standard** or **Voigt** notation. When **User defined** is selected, a 6-by-6 symmetric matrix is displayed to enter the coefficients of $\mathbf{A}$.

Extended Tube

From the **Compressibility** list select how the material is specified in terms of the strain energy density — **Compressible, uncoupled**; **Nearly incompressible**; or **Incompressible**.

- If the **Compressible, uncoupled** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or **User defined**. The **Bulk modulus** K, and for the **Polynomial** option the parameters K_2 and K_3, use values **From material**.
- If the **Nearly incompressible** option is selected, specify the **Volumetric strain energy density** — **Quadratic**, **Polynomial**, **Logarithmic**, **Hartmann–Neff**, **Miehe**, **Simo–Taylor**, or

User defined. The **pressure formulation** is selected from the **Use mixed formulation** list, and the default value for the **Bulk modulus** κ is 100 times the equivalent shear modulus.

- If the **Incompressible** option is selected, an extra variable and weak constraint is added to enforce the incompressibility condition $J_{el} = 1$.

The **Model parameters** G_c, G_e, α, and β use values **From material**.

User Defined

If a **Compressible** material is selected from the **Compressibility** list, enter an expression for the **Elastic strain energy density** W_s.

You can also use a mixed formulation by adding the mean pressure as an extra dependent variable. In this case, select either **Nearly incompressible** or **Incompressible** from the **Compressibility** list.

If **Nearly incompressible** is selected, enter the **Isochoric strain energy density** W_{iso} and the **Volumetric strain energy density** W_{vol}.

If **Incompressible** is selected, enter the **Isochoric strain energy density** W_{iso} only. An extra weak constraint is added to enforce the incompressibility condition $J_{el} = 1$.

Select the **Use elastic deformation gradient** checkbox to compute the strain energy densities based on components of the elastic deformation gradient F_{el}. The checkbox is not selected by default, so it is assumed that W_s, W_{iso}, and W_{vol} are expressions of the components or invariants of the elastic right Cauchy–Green tensor C_{el}.

For examples of:

- **Mooney–Rivlin, two-parameters** and **Ogden**, see *Inflation of a Spherical Rubber Balloon*. Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/balloon_inflation**.
- **Murnaghan**, see *Elasto-Acoustic Effect in Rail Steel*. Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/rail_steel**.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

DISCRETIZATION

If the hyperelastic material is nearly incompressible or incompressible, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

The **Discretization** section is available when you use mixed formulation. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

ESTIMATE OF MAXIMUM WAVE SPEED

This section is only present in the Solid Mechanics, Explicit Dynamics interface.

Select how the maximum wave speed in the material is determined — **Automatic** or **User defined**. For **User defined**, enter the maximum wave speed, $c_{\max}$.

ADVANCED

This section is only displayed when **User defined** has been selected as the material model.

Enter the **Equivalent Young's modulus** E_{eq} and the **Equivalent shear modulus** G_{eq}. The defaults are `1 GPa` and $E_{\text{eq}}/3$, respectively. The equivalent moduli are defined by most hyperelastic material models, but not in the **User defined** option. The characteristic stiffness is needed in expressions for the default penalty factors in contact methods, and it should be representative for the stiffness of the destination domain material in a direction normal to the boundary.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Hyperelastic Material**
Solid Mechanics, Explicit Dynamics > **Material Models** > **Hyperelastic Material**
Layered Shell > **Material Models** > **Hyperelastic Material**
Shell > **Material Models** > **Hyperelastic Material, Layered**
Membrane > **Material Models** > **Hyperelastic Material**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Material Models** > **Hyperelastic Material**

Physics tab with **Shell**, **Layered Shell** or **Membrane** selected:

Boundaries > **Material Models** > **Hyperelastic Material**

Shape Memory Alloy

The **Shape Memory Alloy** feature is used to model stress–strain relationships that are nonlinear even at infinitesimal strains. This material model requires the Nonlinear Structural Materials Module. **Shape Memory Alloy** is available for 3D, 2D, and 2D axisymmetry.

By adding the following subnodes to the **Shape Memory Alloy** node you can incorporate other effects:

- Phase Transformation Direction
- Thermal Expansion (for Materials)
- Initial Stress and Strain
- External Stress
- External Strain
- Damping

Note: Some options are only available with certain COMSOL products (see www.comsol.com/products/specifications/).

When a **Shape Memory Alloy** node is present, two result templates are added. In **Martensite Volume Fraction**, the distribution of martensite over the domain is shown. The **Shape Memory Alloy Phase Diagram** plot contains a graph showing the material properties in terms of a phase diagram.

See also Shape Memory Alloy in the *Structural Mechanics Theory* chapter.

MODEL INPUTS

From the **Temperature** *T* list, select an existing temperature variable from a heat transfer interface (for example, **Temperature (ht)**), if any temperature variables exist, or select **User defined** to enter a value or an expression for the temperature.

If any material in the model has a temperature dependent mass density, and **From material** is selected for the density, the **Volume reference temperature** list will appear in the **Model Input** section. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

SHAPE MEMORY ALLOY

Select a **Shape memory alloy model** from the list: **Lagoudas** or **Souza–Auricchio**.

Lagoudas

For **Lagoudas**, the **Reference temperature** T_0, the **Poisson's ratio** ν, and the **Density** ρ are taken **From material**. For **User defined** enter other values or expressions.

For **Austenite**, select a material from the list. The **Young's modulus** E_A and the **Heat capacity at constant pressure** $C_{p,A}$ are taken from the selected material. For **Martensite**, select a material from the list. The **Young's modulus** E_M and the **Heat capacity at constant pressure** $C_{p,M}$ are taken from the selected material. For **User defined** enter other values or expressions.

Under **Phase transformation**, specify the parameters that describe the phase transitions in terms of **Temperature** or **Stress** levels.

- When **Temperature** is selected from the **Specify** list, enter the **Martensite start temperature** M_s, the **Martensite finish temperature** M_f, the **Slope of martensite limit curve** C_M, the **Austenite start temperature** A_s, the **Austenite finish temperature** A_f, and the **Slope of austenite limit curve** C_A.
- When **Stress** is selected from the **Specify** list, enter the **Martensite start stress** σ_{Ms}, the **Martensite finish stress** σ_{Mf}, the **Slope of martensite limit curve** C_M, the **Austenite start stress** σ_{As}, the **Austenite finish stress** σ_{Af}, the **Slope of austenite limit curve** C_A, and the **Measurement temperature** T_σ.

Under the **Maximum transformation strain** list select **Constant** to directly enter the **Maximum transformation strain** $\varepsilon_{tr,max}$, or **Exponential law** to specify a stress-dependent maximum transformation strain. Under **Exponential law**, enter the **Initial maximum transformation strain** $\varepsilon_{tr,min}$, the **Ultimate transformation strain** $\varepsilon_{tr,sat}$, the **Critical stress** σ_{crit}, and the **Saturation exponent** k. Enter the **Calibration stress level** σ^*.

Under **Phase transformation kinetics**, select the **Transformation function** from the list: **Quadratic**, **Cosine**, **Smooth**, or **User defined**.

For **Smooth**, enter the smoothing parameters η_1, η_2, η_3, and η_4.

For **User defined** enter the **Yield stress** σ_{ys}, the **Forward transformation law**, and the **Reverse transformation law**.

When **Lagoudas** model is selected, a **Phase Transformation Direction** subnode is added to the **Shape memory alloy** node. Select a **Transformation direction** from the list: **Automatic** (default) or **User defined**.

Souza–Auricchio

For **Souza–Auricchio** the defaults for the **Poisson's ratio** ν and **Density** ρ, are taken **From material**. For **User defined** enter other values or expressions.

For **Austenite**, select a material from the list. The **Young's modulus** E_A is taken from the selected material. For **Martensite**, select a material from the list. The **Young's modulus** E_M is taken from the selected material. For **User defined** enter other values or expressions.

Under **Phase transformation** specify how the **Equivalent stress** is computed. Select **von Mises** for a symmetric elastic domain radius in tension or compression. Select **Prager–Lode** to specify the **Stress ratio** between the elastic domain radius in compression and tension.

The default values for the **Slope of limit curve** β and the **Maximum transformation strain** $\varepsilon_{tr,max}$ are taken **From material**. For **User defined** enter other values or expressions.

Specify the parameters that describe the phase transitions in terms of **Temperature** or **Stress** levels.

- When **Standard** is selected from the **Specify** list, enter the **Elastic domain radius** σ_0, the **Hardening modulus** H_k, and the **Reference temperature** T^*.
- When **Stress** is selected from the **Specify** list, enter the **Martensite start stress** σ_{Ms}, the **Martensite finish stress** σ_{Mf}, the **Austenite finish stress** σ_{Af}, and the **Measurement temperature** T_σ.
- When **Temperature** is selected from the **Specify** list, enter the **Martensite start temperature** M_s, the **Martensite finish temperature** M_f, and the **Austenite finish temperature** A_f.

Initial Transformation State

For **Lagoudas** model, enter the **Initial martensite volume fraction**, the **Initial transformation strain tensor**, the **Initial martensite volume fraction at reverse point**, and the **Initial transformation strain tensor at reverse point**.

For the **Souza–Auricchio** model, enter the **Initial transformation strain tensor**.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

Only the additive decomposition of strains is available for Lagoudas model.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Local method** to solve the plasticity problem — **Automatic** or **Backward Euler**. When **Backward Euler** is selected, it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

See also the Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

For the Lagoudas model, under **Volume fraction constraint** enter the **Penalty factor** γ to constrain the *martensite volume fraction* by the inequality $\xi_M - 1 < 0$.

For the Souza–Auricchio model, under **Transformation strain constraint** enter the **Penalty factor** γ to constrain the *equivalent transformation strain* by the inequality $\varepsilon_{tre} - \varepsilon_{tre,max} < 0$.

Check the **Use transition zone** checkbox to specify a transition zone size for the inequality.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Shape Memory Alloy**
Truss > **Material Models** > **Shape Memory Alloy**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Shape Memory Alloy**

Physics tab with **Truss** selected:

Edges > **Material Models** > **Shape Memory Alloy**

Phase Transformation Direction

When **Lagoudas** is selected from the **Shape memory alloy model** list, the **Phase Transformation Direction** node is automatically added to the **Shape Memory Alloy** node. Since in many applications the transformation direction $\dot{\xi}$ is known a priori (for instance, mechanical loading or unloading, or temperature increment/decrement) a user input enables to set the transformation direction manually to 1 for forward phase transformation from austenite to martensite, $\dot{\xi} > 0$, or -1 for phase transformation from martensite to austenite, $\dot{\xi} < 0$, thus speeding up the computational time.

PHASE TRANSFORMATION DIRECTION

Select a **Transformation Direction** — **Automatic** or **User Defined**.

- For an example of a user defined **Phase Transformation Direction**, see *Shape Memory Alloy Self-Expanding Stent*: Application Library path **Nonlinear_Structural_Materials_Module/Shape_Memory_Alloys/sma_stent**.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Shape Memory Alloy** > **Phase Transformation Direction**
Truss > **Shape Memory Alloy** > **Phase Transformation Direction**

Ribbon

Physics tab with **Shape Memory Alloy** node selected in the model tree:

AttributesPhase Transformation Direction

Piezoelectric Material

The **Piezoelectric Material** node defines the piezoelectric material properties either in stress–charge form using the elasticity matrix and the coupling matrix, or in strain–charge form using the compliance matrix and the coupling matrix. It is normally used together with a **Piezoelectricity** multiphysics coupling node and a corresponding **Charge Conservation, Piezoelectric** node in the **Electrostatics** interface. This node is added by default to the **Solid Mechanics** interface when adding a **Piezoelectricity** interface. **Piezoelectric Material** available for 3D, 2D, and 2D axisymmetry.

This material model requires the Structural Mechanics Module, or MEMS Module, or Acoustics Module.

By adding the following subnodes to the **Piezoelectric Material** node you can incorporate many other effects:

- Initial Stress and Strain
- Thermal Expansion (for Materials)
- Mechanical Damping
- Coupling Loss
- Dielectric Loss
- Conduction Loss (Time-Harmonic)

When the **Piezoelectric Material** node is added to the Solid Mechanics interface in the absence of an active **Piezoelectricity** multiphysics coupling node, the material behaves similarly to a Linear Elastic Material node. The elastic properties correspond to the elasticity or compliance matrix entered (see below). The piezoelectric effect is then *not* included in the equation system.

See also Piezoelectricity in the *Structural Mechanics Theory* chapter.

The **Piezoelectric Material** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

PIEZOELECTRIC MATERIAL PROPERTIES

Select a **Constitutive relation** — **Stress–charge form** or **Strain–charge form**. For each of the following, the default uses values **From material**. For **User defined** enter other values in the matrix or field as needed.

- For **Stress–charge form**, select an **Elasticity matrix, Voigt notation** (c_E).

- For a **Strain–charge form**, select a **Compliance matrix, Voigt notation** (s_E).
- Select a **Coupling matrix, Voigt notation** (e_{ES} or d_{ET}).
- Select a **Relative permittivity** (ε_{rS} or ε_{rT}).
- Enter values for the **Remanent electric displacement** ($\mathbf{D}_r$).
- Select a **Density** (ρ).

For entering these matrices, use the following order (Voigt notation), which is the common convention for piezoelectric materials: *xx*, *yy*, *zz*, *yz*, *xz*, *zy*.

Check the **Use multiplicative formulation** checkbox to use a formulation based on the multiplicative decomposition of elastic and inelastic (piezoelectric) strains.

When the **Use multiplicative formulation** checkbox is selected, all studies in the model become geometrically nonlinear. The **Include geometric nonlinearity** checkbox on the study step **Settings** window is selected and cannot be cleared.

See also Multiplicative Formulation for Piezoelectricity in the *Structural Mechanics Theory* chapter.

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which adds an extra dependent variable for either the pressure or for the volumetric strain. For details, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, or **Strain formulation**.

Density

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can

also select **User defined** to enter a value or an expression for the reference temperature locally.

The density is needed for dynamic analysis and when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input.
- Modeling Piezoelectric Problems in the *Structural Mechanics Modeling* chapter.
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced

integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations.

When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Piezoelectric Material**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Piezoelectric Material**

Piezomagnetic Material

The **Piezomagnetic Material** node defines the linear magnetoelastic material properties. The material data can be entered either in the strain–magnetization form using the elasticity matrix and the coupling matrix, or in stress–magnetization form using the compliance matrix and the coupling matrix. It is normally used as part of Piezomagnetism multiphysics interface together with a **Piezomagnetism** multiphysics coupling node and **Ampère's Law, Piezomagnetic** node in the corresponding **Magnetic Fields** interface. **Piezomagnetic Material** node is added by default to the **Solid Mechanics**

interface when adding a Piezomagnetism multiphysics interface. This material model available for 3D, 2D, and 2D axisymmetry.

When the **Piezomagnetic Material** node is added to the Structural Mechanics interface in the absence of an active **Piezomagnetism** multiphysics coupling node, the material behaves similarly to a Linear Elastic Material node with some limitations on the format for the elastic material data input. All the magnetic material data and coupling data will have no effect. The piezomagnetic effect is then *not* included in the corresponding equation system.

By adding the following subnodes to the **Piezomagnetic Material** node you can incorporate other effects:

- Initial Stress and Strain
- Thermal Expansion (for Materials)
- Mechanical Damping

See also Magnetostriction and Piezomagnetism in the *Structural Mechanics Theory* chapter.

The **Piezomagnetic** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

PIEZOMAGNETIC MATERIAL PROPERTIES

Select a **Constitutive relation** — **Strain–magnetization form** or **Stress–magnetization form**. For each of the following, the default uses values **From material**. For **User defined**, enter other values in the matrix or field as needed.

- For **Strain–magnetization form**, select a **Compliance matrix, Voigt notation** (s_{H}).
- For a **Stress–magnetization form**, select an **Elasticity matrix, Voigt notation** (c_{H}).
- Select a **Piezomagnetic coupling matrix, Voigt notation** (d_{HT} or e_{HS}).
- Select a **Relative permeability** (μ_{rT} or μ_{rS}).
- Select a **Density** (ρ).

For entering these matrices, use the following order (Voigt notation), which is the common convention for piezomagnetic materials: *xx*, *yy*, *zz*, *yz*, *xz*, *zy*.

Check the **Use multiplicative formulation** checkbox to use a formulation based on the multiplicative decomposition of elastic and inelastic (piezomagnetic) strains.

When the **Use multiplicative formulation** checkbox is selected, all studies in the model become geometrically nonlinear. The **Include geometric nonlinearity** checkbox on the study step **Settings** window is selected and cannot be cleared.

- See also Multiplicative Formulation for Piezomagnetism in the *Structural Mechanics Theory* chapter.
- Modeling Magnetostrictive Materials in the *Structural Mechanics Modeling* chapter.
- Piezomagnetism in the *Multiphysics Interfaces and Couplings* chapter.
- Ampère's Law, Piezomagnetic in the *AC/DC Module User's Guide*.
- The Magnetic Fields Interface in the *COMSOL Multiphysics Reference Manual*.

Density

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input**

section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced

integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations.

When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **Piezomagnetic Material**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Piezomagnetic Material**

Magnetostrictive Material

The **Magnetostrictive Material** node is obsolete and will be removed in future versions. It cannot be added in version 6.1 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use one of the new multiphysics interfaces instead, either The Nonlinear Magnetostriction Interface or The Piezomagnetism Interface (for modeling linear magnetostrictive effects).

Viscoelasticity

Use the **Viscoelasticity** subnode to add viscous stress contributions to an elastic material model. This material model is available in the Solid Mechanics, Shell, Layered Shell, and Membrane interfaces, and can be used together with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Hyperelastic Material**, **Linear Elastic Material, Layered**, and **Hyperelastic Material, Layered**.

Note: Some options described below are only available with certain COMSOL products (see www.comsol.com/products/specifications/).

See also Linear Viscoelasticity and Large Strain Viscoelasticity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is only present when **Viscoelasticity** is used as a subnode to:

- A material model in the Layered Shell interface. See the documentation for the Viscoelasticity node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** or **Layered Hyperelastic Elastic Material** in the Shell interface. See the documentation for the Viscoelasticity node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Viscoelasticity node in the Membrane chapter.

SKETCH

Viscoelastic models are often conceptually viewed as rheological networks consisting of spring and damper elements. The sketch section shows the arrangement of springs and dampers for the chosen viscoelastic material model. If fractional derivatives are enabled, the dampers are replaced by so-called spring-pot elements. Relevant input parameters are shown next to the spring, damper, and spring-pot elements. The elastic stiffness contribution of the parent material model is indicated by a light gray spring symbol representing the bulk and shear moduli, K and G, respectively.

VISCOELASTICITY MODEL

Select a **Material model** — **Generalized Maxwell**, **Generalized Kelvin–Voigt**, **Maxwell**, **Kelvin–Voigt**, **Standard linear solid**, **Burgers**, or **User defined**. Then, enter the settings for each option that follows.

From the **Viscoelastic strains** list select **Volumetric**, **Deviatoric**, or **Volumetric and deviatoric**. Select **Volumetric** when the viscoelastic behavior applies only to the volumetric deformation. The **Deviatoric** option (default) applies the viscoelastic relaxation to the shear deformation only. With **Volumetric and deviatoric** the viscoelastic strain is full.

For some material models, you can select the stiffness to use when solving a stationary problem. Select the **Stiffness used in stationary studies** — **Long-term** or **Instantaneous**. With **Long-term** all dampers are assumed to be fully relaxed, whereas with **Instantaneous** all dampers are assumed to be rigid.

Generalized Maxwell

For **Generalized Maxwell** enter the values for the parameters that describe the viscoelastic behavior as a series of spring-dashpot pairs.

Depending on the selection done in the **Viscoelastic strains** list, for each **Branch** row enter the stiffness of the spring K_m in the **Bulk modulus (Pa)** column and/or G_m in the **Shear modulus (Pa)** column, and the relaxation time constant τ_m in the **Relaxation time (s)** column for the spring-dashpot pair in branch m.

When the **Use fractional derivatives** checkbox is selected, enter the fractional order β_m in the **Fractional order (1)** column for each spring-spring-pot branch.

For large strain viscoelasticity, in each **Branch** row enter the energy factor of the branch, β_{vm}, in the **Energy factor (1)** column and the relaxation time constant τ_m in the **Relaxation time (s)** column for the spring-dashpot pair.

- Use the **Add** button (+) to add a row to the table, the **Delete** button ( ) to delete a row in the table, or the **Clear Table** button ( ) to clear the whole table.
- Use the **Load from file** button ( ) and the **Save to file** button ( ) to load and store data for the branches in a text file with space-separated columns.

When the **Prune viscoelastic branches** checkbox is selected, enter the **Cutoff frequencies** f_{lower} and f_{upper}. The relaxation times τ_m are frozen and cannot be changed when the checkbox is selected. In order to change these settings again, clear the checkbox because pruning is only performed at the time when the checkbox is selected. It is also required that the relaxation times for each branch have constant values when **Prune viscoelastic branches** is selected.

From the **Stiffness used in stationary studies** list, select either **Long-term** or **Instantaneous**. With **Long-term**, all dampers are assumed to be relaxed; hence the branches do not contribute to the stress. The material stiffness is therefore given by the stiffness in the parent material model (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material). With **Instantaneous**, all dampers are assumed to be rigid, and the material stiffness is given by springs arranged in parallel.

Generalized Kelvin–Voigt

For **Generalized Kelvin–Voigt** enter the values for the parameters that describe the viscoelastic behavior of multiple Kelvin–Voigt elements arranged in series.

Depending on the selection done in the **Viscoelastic strains** list, for each **Branch** row enter the stiffness of the spring K_m in the column labeled **Bulk modulus (Pa)** and/or G_m in the column labeled **Shear modulus (Pa)**, and the relaxation time τ_m in the column labeled **Relaxation time (s)** for the spring-dashpot pair in the element m.

When the **Use fractional derivatives** checkbox is selected, enter the fractional order β_m in the **Fractional order (1)** column for each spring-spring-pot branch.

- Use the **Add** button (+) to add a row to the table, the **Delete** button ( ) to delete a row in the table, or the **Clear Table** button ( ) to clear the whole table.
- Use the **Load from file** button ( ) and the **Save to file** button ( ) to load and store data for the elements in a text file with space-separated columns.

Select the **Stiffness used in stationary studies**, either **Long-term** or **Instantaneous**. With **Long-term**, all dampers are assumed to be relaxed. The material stiffness is therefore

given by springs arranged in series. With **Instantaneous**, all dampers are assumed to be rigid; hence the viscoelastic branches do not contribution to the strain, and the instantaneous stiffness is determined by the parent material only (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material).

Maxwell

For **Maxwell** enter the parameters that describes the viscous behavior of a single dashpot connected in series with a spring.

Depending on the selection done in the **Viscoelastic strains** list, the relaxation time or viscosity is applied to the volumetric, deviatoric, or both volumetric and deviatoric deformation. Select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time** τ_v. The default is 1 ms.
- **Viscosity** η_v of the dashpot. The default is 1000 Pa·s.

When the **Use fractional derivatives** checkbox is selected, enter the fractional order β_v of the spring-pot. The default is 0.8 (dimensionless).

Note that the instantaneous stiffness is given by the parent material model (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material).

Kelvin–Voigt

For **Kelvin–Voigt** enter the values for the parameter that describes the viscous behavior of the single dashpot in parallel with a spring.

Depending on the selection done in the **Viscoelastic strains** list, the relaxation time or viscosity is applied to the volumetric, deviatoric, or both volumetric and deviatoric deformation. Select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time** τ_v of the dashpot. The default is 1 ms.
- **Viscosity** η_v of the dashpot. The default is 1000 Pa·s.

When the **Use fractional derivatives** checkbox is selected, enter the fractional order β_v of the spring-pot. The default is 0.8 (dimensionless).

Note that the instantaneous stiffness is given by the parent material model (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material).

Standard Linear Solid

For **Standard linear solid** enter the values for the parameters that describe the viscoelastic behavior of the single spring-dashpot branch.

Depending on the selection done in the **Viscoelastic strains** list, enter the **Bulk modulus** and/or the **Shear modulus** of the spring in the K_v and G_v fields. The default values are 10 MPa and 1 MPa, respectively.

Note that the long-term stiffness is given by the parent material model (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material).

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time** τ_v of the dashpot. The default is 1 ms.
- **Viscosity** η_v of the dashpot. The default is 1000 Pa·s.

For large strain viscoelasticity, enter the **Energy factor** β_v of the dashpot. The default is 0.2 (dimensionless).

When the **Use fractional derivatives** checkbox is selected, enter the fractional order β_v of the spring-dashpot. The default is 0.8 (dimensionless). When this checkbox is selected, an extra section **Time Domain and Eigenfrequency** becomes available. There, you can compute an approximation for the complex valued fractional elasticity modulus contribution from the spring-dashpot branch. By default, the approximation is not computed. Without it, the fractional viscoelasticity model is not applicable for computations using a Time Dependent study step, and the eigenvalue problem will become nonlinear for an Eigenfrequency study step.

The approximation computation needs to be done as a preprocessing step. Use the **Compute approximation** ( ) action button that is available at the **Time Domain and Eigenfrequency** section. Once the computation has been performed, you can preview the result using the **Preview approximation** ( ) action button that will become active at the section. The approximation corresponds to the Generalized Maxwell viscoelasticity model without fraction derivatives. You can check the Show **approximation data** checkbox to inspect the computed parameter values. Thus

computed approximation can be used in Eigenfrequency and Time Dependent study steps. For the eigenfrequency computations, the eigenvalues problem will be linear.

The **Viscoelasticity** subnode is contributing in a sense that stress contributions from two or more such subnodes added under the same parent node are additive.

You can add several **Viscoelasticity** subnodes and set them to use **Standard linear solid** model and **Use fractional derivatives** option and compute the approximation for each node. Such set-up will be an effective equivalent of the Generalized Maxwell model with fractional derivatives, which can be used in Eigenfrequency and Time Dependent study steps.

Burgers

For **Burgers** enter the values for the parameter that describes the viscous behavior of the spring dashpot in series with a second spring-dashpot pair.

Depending on the selection done in the **Viscoelastic strains** list, enter the **Bulk modulus** and/or the **Shear modulus** of the second spring in the K_{v2} and G_{v2} fields. The default values are 1 MPa.

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time**. The defaults are 1 ms and 10 ms for τ_{v1} and τ_{v2}, respectively.
- **Viscosity**. Enter the viscosity of the dashpots. The defaults are 1000 Pa·s and 100 Pa·s for η_{v1} and η_{v2}, respectively.

When the **Use fractional derivatives** checkbox is selected, enter the fractional orders of the spring-pot pairs. The defaults are 0.8 and 0.9 (dimensionless) for β_{v1} and β_{v2}, respectively.

Note that the instantaneous stiffness is given by the parent material model (for example, Linear Elastic Material, Nonlinear Elastic Material or Hyperelastic Material).

User Defined

When **Volumetric** is selected from the **Viscoelastic strains** list, specify contributions to either the **Storage and loss moduli** K' and K'', or the **Storage and loss compliances** Q' and Q'', or the **Loss factor** η_v that defines the complex–valued bulk modulus.

When **Deviatoric** is selected from the **Viscoelastic strains** list, specify contributions to either the **Storage and loss moduli** G' and G'', or the **Storage and loss compliances** J' and J'', or the **Loss factor** η_v that defines the complex–valued shear modulus.

When **Volumetric and deviatoric** is selected from the **Viscoelastic strains** list, specify contributions to either the **Storage and loss moduli** K', K'', G' and G'', or the **Storage and loss compliances** Q', Q'', J' and J'', or the **Loss factor** η_v that defines the complex–valued bulk and shear moduli.

The expressions can be entered as functions taken directly from interpolated data, or can be analytical expressions of the frequency variable.

The internal variables for the frequency f and angular frequency ω are named *phys*`.freq` and *phys*`.omega`, respectively. Here, *phys* is the tag of the parent physics (for instance, `solid`). You can also use the variable `freq` defined by the solver.

At the **Time Domain and Eigenfrequency** section, you can compute an approximation for the complex valued fractional elasticity modulus contribution from the spring-dashpot branch. By default, the approximation is not computed. Without it, the **User defined** model is not applicable for computations using a Time Dependent study step, and the eigenvalue problem can become nonlinear for an Eigenfrequency study step.

The approximation computation needs to be done as a preprocessing step. Use the **Compute approximation** ( ) action button that is available at the **Time Domain and Eigenfrequency** section. Once the computation has been performed, you can preview it using the **Preview plot** ( ) action button that will become active at the section. The approximation corresponds to the Generalized Maxwell viscoelasticity model without fraction derivatives. You can check the **Show approximation data** checkbox to inspect the computed parameter values. The thus computed approximation will be used in

Eigenfrequency and Time Dependent study steps. For the eigenfrequency computations, the eigenvalues problem will be always linear.

The **Viscoelasticity** subnode is contributing in a sense that stress contributions from two or more such subnodes added under the same parent node are additive.

You can add several **Viscoelasticity** subnodes and set them to use **User defined** model to build up an effective multi branch viscoelastic model of any degree of complexity, which can be configured for use in Eigenfrequency and Time Dependent study steps.

THERMAL EFFECTS

Viscoelastic properties have a strong dependence on the temperature. For thermorheologically simple materials, a change in the temperature can be transformed directly into a change in the time scale. Thus, the relaxation time is modified to $a_T(T)\tau_m$, where $a_T(T)$ is a shift function.

Select a Shift function — **None**, **Williams–Landel–Ferry**, **Arrhenius**, **Tool–Narayanaswamy–Moynihan**, or **User defined**.

- When the default, **None**, is kept, the shift function $a_T(T)$ is set to unity and the relaxation time is not modified.
- For **Williams–Landel–Ferry** enter values or expressions for these properties:
  - **Reference temperature** T_{ref}. The default is 293.15 K.
  - **WLF constant 1** $C_{1\text{WLF}}$. The default is 17.44.
  - **WLF constant 2** $C_{2\text{WLF}}$. The default is 51.6 K.
- For **Arrhenius** enter values or expressions for these properties:
  - **Reference temperature** T_{ref}. The default is 293.15 K.
  - **Activation energy** Q.

- For **Tool–Narayanaswamy–Moynihan** enter values or expressions for these properties:
  - **Reference temperature** T_{ref}. The default is 293.15 K.
  - **Activation energy** Q.
  - **Activation energy fraction** χ ($0<\chi<1$).
  - **Structural relaxation times** λ_0.
  - **Fictive temperature weights** w.
- For **User defined** enter a value or an expression for the shift function a_T.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The **Use local time integration** checkbox is selected by default. Clear it in case you want to use the global time integration scheme. The checkbox is only available for the Generalized Maxwell and Standard Linear Solid models. For all other viscoelasticity models, the global time integration is used.

Clear the **Use local time integration** checkbox to select the **Shape function type** — **Discontinuous Lagrange** (default) or **Gauss-point data** for the components of the auxiliary viscoelastic tensor. When the discontinuous Lagrange discretization is used, the shape function order is set as one order lower than the order used for the displacement field. This results fewer degrees of freedom being added to the model than when using Gauss-point data. The accuracy does in general not differ much. If you want to enforce that the constitutive law is fulfilled at the integration points, select **Gauss-point data**.

To compute the energy dissipation caused by viscoelasticity, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

- *Viscoelastic Structural Damper*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_frequency**
- *Eigenmodes of a Viscoelastic Structural Damper*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_eigenmodes**
- *Viscoelastic Structural Damper — Transient Analysis*: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/viscoelastic_damper_transient**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Viscoelasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Viscoelasticity**
Solid Mechanics > **Hyperelastic Material** > **Viscoelasticity**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Viscoelasticity**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Viscoelasticity**
Shell > **Linear Elastic Material, Layered** > **Viscoelasticity**
Shell > **Hyperelastic Material, Layered** > **Viscoelasticity**
Layered Shell > **Linear Elastic Material** > **Viscoelasticity**
Layered Shell > **Hyperelastic Material** > **Viscoelasticity**
Membrane > **Linear Elastic Material** > **Viscoelasticity**
Membrane > **Linear Elastic Material, Layered** > **Viscoelasticity**
Membrane > **Nonlinear Elastic Material** > **Viscoelasticity**
Membrane > **Hyperelastic Material** > **Viscoelasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Hyperelastic Material**, **Linear Elastic Material, Layered**, or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Viscoelasticity**

Mullins Effect

Use the **Mullins effect** subnode to define the properties for modeling the stress-softening phenomenon under cyclic loading. The Mullins effect can be used together with Hyperelastic Material.

SHELL PROPERTIES

This section is only present when **Mullins effect** is used as a subnode to **Hyperelastic Material** is used in the Layered Shell interface. See the documentation for the Mullins Effect node in the Layered Shell chapter.

MULLINS EFFECT

The **Damage function** defines the model for Mullins effect. Select **Ogden–Roxburgh** or **Miehe**.

For **Ogden–Roxburgh** enter the following settings:

- **Maximum damage** d_∞. The default is 1.
- **Damage saturation** W_{sat}. The default is 1 MJ/m^3.
- **Deformation dependence coefficient** α. The default is 0.
- Select **Use error function** checkbox to define the damage evolution with error function instead of the hyperbolic function (default).

For **Miehe** enter the following settings:

- **Maximum damage** d_∞. The default is 1.
- **Damage saturation** W_{sat}. The default is 1 MJ/m^3.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Hyperelastic Material** > **Mullins Effect**
Shell > **Hyperelastic Material, Layered** > **Mullins Effect**
Layered Shell > **Hyperelastic Material** > **Mullins Effect**
Membrane > **Hyperelastic Material** > **Mullins Effect**

Ribbon

Physics tab with **Hyperelastic Material** or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Mullins Effect**

Plasticity

Use the **Plasticity** subnode to define the properties for modeling elastoplastic materials. This material model is available in the Solid Mechanics, Shell, Layered Shell,

Membrane, and Truss interfaces, and can be used together with Linear Elastic Material, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module or the Geomechanics Module are required for this material model, and the available options depend on the used products. For details, see www.comsol.com/products/specifications/.

When a **Plasticity** node is present, an **Equivalent Plastic Strain** plot is available under **Result Templates**.

See also Elastoplastic Materials in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is present when **Plasticity** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Plasticity node in the Layered Shell chapter.
- **Hyperelastic Material** in the Layered Shell interface. See the documentation for the Plasticity node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Plasticity node in the Shell and Plate chapter.
- **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Plasticity node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Plasticity node in the Membrane chapter.

PLASTICITY MODEL

Use this section to define the plastic properties of the material.

Equivalent Stress

The yield function defines the limit of the elastic regime, $F(\sigma_e, \sigma_{ys}) \leq 0$, and sets the onset for plastic deformation. Changing the equivalent stress measure σ_e allows to specify different yield criteria. See Yield Criterion for details.

Select the **Equivalent stress** — **von Mises**, **Tresca**, **Hill orthotropic**, or **User defined** to define the yield criterion.

- The default is **von Mises** criterion with associate plastic potential.
- Select **Tresca** to use Tresca yield criterion. The plastic potential can be an **Associated** or nonassociated flow rule with the **von Mises** stress as plastic potential.
- Select **Hill orthotropic** to use Hill's criterion. From the **Specify** list select the **Initial tensile and shear yield stresses** σ_{ys0ij}, **Hill's coefficients** F, G, H, L, M, and N, or **Hill's coefficients, dimensionless** f, g, h, l, m, and n. The default uses values **From material** (if it exists) or **User defined**. See Expressions for Hill's Coefficients for details. The principal directions of orthotropy are inherited from the coordinate system selection in the parent node.
- For **User defined** enter a value or an expression for the equivalent stress. Write any expression in terms of the stress tensor components or its invariants in the $\phi(\sigma)$ field.

Initial Yield Stress

- The default **Initial yield stress** σ_{ys0} uses values **From material**. It represents the stress level where plastic deformation starts. For Hill's criterion, it is also possible to select the **Reciprocal average** yield stress, see Expressions for Hill's Coefficients for details.

Plastic Potential

Select the **Plastic potential** Q_p related to the flow rule — **Associated, von Mises**, or **User defined** (nonassociated). Enter a **User defined** value in the Q_p field as needed.

Equivalent Plastic Strain

Select how the **Equivalent plastic strain** ε_{pe} is computed — **Associated**, **von Mises**, or **User defined**. Enter a **User defined** value in the h_p field as needed. See Hardening Rule for details.

Isotropic Hardening

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list. See Isotropic Hardening for details.

- Select **Perfectly plastic** (ideal plasticity) if the material can undergo plastic deformation without any increase in yield stress.
- For **Linear** the default **Isotropic tangent modulus** $E_{T\text{iso}}$ uses values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified as hardening occurs, and it is related to the equivalent plastic strain ε_{pe} as

$$\sigma_{ys} = \sigma_{ys0} + E_{iso}\varepsilon_{pe} \text{ with } \frac{1}{E_{iso}} = \frac{1}{E_{Tiso}} - \frac{1}{E}$$

For the linear isotropic hardening model, the yield stress increases proportionally to the equivalent plastic strain ε_{pe}. The Young's modulus E is taken from the elastic material properties.

- Select **Ludwik** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the power law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + k\varepsilon_{pe}^{n}$$

The **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.

- Select **Johnson–Cook** or **Modified Johnson–Cook** from the list to model strain rate dependent hardening. The **Strength coefficient** k, **Hardening exponent** n, **Reference strain rate** $\dot{\varepsilon}_0$, and **Strain rate strength coefficient** C use values **From material** (if it exists) or **User defined**.

  Select a **Thermal softening model** — **No thermal softening**, **Power law**, or **User defined**.

  - For **Power law**, enter the **Reference temperature** T_{ref}, the **Melting temperature** T_m, and the **Temperature exponent**, m.
  - For **User defined**, enter the **Thermal Softening function** $f(T_h)$, the **Reference temperature** T_{ref}, and the **Melting temperature** T_m. The softening function $f(T_h)$ typically depends on the built-in variable for the normalized homologous temperature T_h and have the properties $f(0) = 0$ and $f(1) = 1$. The variable is named using the scheme `<physics>.<elasticTag>.<plasticTag>.Th`, for example, `solid.lemm1.plsty1.Th`.

The yield stress and hardening function for the Johnson–Cook model is given by

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})\left(1 + C\log\left(\frac{\dot{\varepsilon}_{pe}}{\dot{\varepsilon}_0}\right)\right)(1 - f(T_h))$$

$$T_h = \frac{T - T_{ref}}{T_m - T_{ref}}$$

For the modified Johnson–Cook model, the yield stress is instead given by

$$\sigma_{ys}(\varepsilon_{pe}) = (\sigma_{ys0} + k\varepsilon_{pe}^{n})\left(1 + C\log\left(1 + \frac{\dot{\varepsilon}_{pe}}{\dot{\varepsilon}_0}\right)\right)(1 - f(T_h))$$

In the case of power law softening, $f(T_h) = T_h^m$.

- For **Swift** nonlinear isotropic hardening, the **Reference strain** ε_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield stress σ_{ys} depends on the power law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0}\left(1 + \frac{\varepsilon_{pe}}{\varepsilon_0}\right)^n$$

Typically, the reference strain equals the onset of plasticity, $\varepsilon_0 = \sigma_{ys0}/E$.

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{pe}}\right)$$

The **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.

- For **Hockett–Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_∞, the **Saturation coefficient** m, and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + (\sigma_\infty - \sigma_{ys0})\left(1 - e^{-m\varepsilon_{pe}^{n}}\right)$$

- For **Hardening function**, the isotropic **Hardening function** $\sigma_h(\varepsilon_{pe})$ uses values **From material** or **User defined**. The yield level σ_{ys} is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\varepsilon_{pe})$$

  - This definition implies that the hardening function $\sigma_h(\varepsilon_{pe})$ in the Material node must be zero at zero plastic strain. In other words, $\sigma_{ys} = \sigma_{ys0}$ when $\varepsilon_{pe} = 0$. With this option it is possible to enter any nonlinear isotropic hardening curve. The hardening function can depend on more variables than the equivalent plastic strain, for example the temperature. Select **User defined** to enter any function of the equivalent plastic strain ε_{pe}. The variable is named using the scheme `<physics>.epe`, for example, `solid.epe`.

Kinematic Hardening

For all yield criteria, select the type of kinematic hardening model from the **Kinematic hardening model** list. See Kinematic Hardening for details.

- Select **No kinematic hardening** (when either ideal plasticity or an isotropic hardening model is selected as isotropic hardening model) if it is a material that can undergo plastic deformation without a shift in the yield surface.
- If **Linear** is selected as the **Kinematic hardening model**, the default **Kinematic tangent modulus** E_k uses values **From material**. This parameter is used to calculate the back stress σ_b as plasticity occurs:

$$\sigma_b = C_k \cdot \frac{2}{3}\varepsilon_p \text{ with } \frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

  This is Prager's linear kinematic hardening model, so the back stress σ_b is collinear to the plastic strain tensor ε_p.
- If **Armstrong–Frederick** is selected from the list, the default **Kinematic hardening modulus** C_k and **Kinematic hardening parameter** γ_k use values **From material**. These parameters are used to calculate the back stress σ_b from the rate equation

$$\dot{\sigma}_b = \frac{2}{3}C_k\dot{\varepsilon}_p - \gamma_k\dot{\varepsilon}_{pe}\sigma_b$$

  This is Armstrong–Frederick nonlinear kinematic hardening model. Replacing the variable for the back strain instead of the back stress

$$\sigma_b = C_k \cdot \frac{2}{3}\varepsilon_b$$

  gives the rate for the back strain

$$\dot{\varepsilon}_b = \dot{\varepsilon}_p - \gamma_k\dot{\varepsilon}_{pe}\varepsilon_b$$

- When **Chaboche** is selected from the **Kinematic hardening model** list, the default **Kinematic hardening modulus** C_0 uses values **From material**. Add branches as needed to solve N rate equations for the back stresses:

$$\dot{\sigma}_{b,i} = \frac{2}{3}C_i\dot{\varepsilon}_p - \gamma_i\dot{\varepsilon}_{pe}\sigma_{b,i}$$

  or equivalently, solve N rate equations for the back strain such as $\sigma_{b,i} = C_i \cdot \frac{2}{3}\varepsilon_{b,i}$

$$\dot{\varepsilon}_{b,i} = \dot{\varepsilon}_p - \gamma_i \dot{\varepsilon}_{pe} \varepsilon_{b,i}$$

The total back stress σ_b is then computed from the sum of the branches

$$\sigma_b = \frac{2}{3} C_0 \varepsilon_p + \frac{2}{3} \sum_{i=1}^{N} C_i \varepsilon_{b,i}$$

For each **Branch** row, enter C_i (the hardening modulus of the branch i) in the **Hardening modulus (Pa)** column and γ_i (the hardening parameter of the branch i) in the **Hardening parameter (1)** column.

Use the **Add** button (+) and the **Delete** button ( ) to add or delete a row in the table. Use the **Load from file** button ( ) and the **Save to file** button ( ) to load and store data for the branches in a text file with three space-separated columns (from left to right): the branch number, the hardening modulus for that branch, and the hardening parameter for that branch.

To compute the energy dissipation caused by plasticity, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

- *Sheet Metal Forming*: Application Library path **Nonlinear_Structural_Materials_Module/Plasticity/sheet_metal_forming**
- For an example on large strain and nonlocal plasticity, see *Necking of an Elastoplastic Metal Bar*: Application Library path **Nonlinear_Structural_Materials_Module/Plasticity/bar_necking**.
- For an example of strain rate dependent plasticity, see *Tensile Test with Strain Rate Dependent Plasticity*: Application Library path **Nonlinear_Structural_Materials_Module/Plasticity/strain_rate_dependent_plasticity**

NONLOCAL PLASTICITY MODEL

Nonlocal plasticity can be used to facilitate, for example, the modeling of material softening. Typical examples that involve material softening are finite-strain plasticity and soil plasticity. In these situations, standard (local) plasticity calculations reveal a mesh and topology dependence, where a mesh refinement fails to reproduce a

physically sound solution. Nonlocal plasticity adds regularization to the equivalent plastic strain, thereby stabilizing the solution.

The default setting is **None**. Select **Implicit Gradient** to add nonlocal regularization to the equivalent plastic strain. Then, enter values or expressions for:

- **Length scale**, l_{int} — The regularization length scale for the equivalent plastic strain ε_{pe} should ideally be greater than the largest mesh element size.
- **Nonlocal coupling modulus**, H_{nl} — The value represents the penalization of the difference between the local and nonlocal variables. A larger value enforces the equivalent plastic strain ε_{pe} to be closer to the nonlocal equivalent plastic strain $\varepsilon_{\text{pe,nl}}$. A typical value would be around 10% the equivalent shear modulus of the material.

See also Nonlocal Plasticity in the *Structural Mechanics Theory* chapter.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog, and it is available with the **Implicit gradient** nonlocal plasticity model.

Select the shape function for the **Nonlocal equivalent plastic strain** $\varepsilon_{\text{pe,nl}}$ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

When a **Tresca** equivalent stress is used with associative plastic flow, it is possible to control the smoothing of the plastic potential. Select **Smoothing of plastic potential** — **Automatic**, or **Manual**. When **Manual** is selected, enter a value for the smoothing parameter σ_{off}. This parameter controls how much the rounded corners of the smooth plastic potential deviates from the Tresca yield surface. A typical value would be around 2% of the initial yield stress.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** or **Backward Euler, damped** is selected,

it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

When the **Backward Euler, damped** method is selected, the Newton's method is enhanced by line search iterations. Using this method can improve the robustness of the plasticity algorithm when the plastic potential or hardening model are highly nonlinear. When selected, it is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

For some 2D problems, for example when using certain anisotropic materials, the out-of-plane shear components of the plastic strain tensor can become non-zero. By default, these strain components are assumed to be zero. Select **Include out-of-plane strains** to solve for all components of the plastic strain tensor.

See also the Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Plasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Plasticity**
Solid Mechanics > **Hyperelastic Material** > **Plasticity**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Plasticity**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Plasticity**
Shell > **Linear Elastic Material, Layered** > **Plasticity**
Shell > **Hyperelastic Material, Layered** > **Plasticity**
Layered Shell > **Linear Elastic Material** > **Plasticity**
Layered Shell > **Hyperelastic Material** > **Plasticity**
Membrane > **Linear Elastic Material** > **Plasticity**
Membrane > **Linear Elastic Material, Layered** > **Plasticity**
Membrane > **Nonlinear Elastic Material** > **Plasticity**

Truss > **Linear Elastic Material** > **Plasticity**
Truss, Explicit Dynamics > **Linear Elastic Material** > **Plasticity**

Ribbon
Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Plasticity**

Set Variables

Use the **Set Variables** subnode to Plasticity, Pressure-Dependent Plasticity, Creep, Viscoplasticity, Porous Plasticity, Soil Plasticity, Concrete, Rocks, or Elastoplastic Soil Material, in order to reset plasticity variables according to a **Setting condition** that you define. When the **Setting condition** is satisfied, the plasticity variables are reset to the specified values.

SET VARIABLES

Enter the **Setting condition**. This is a Boolean expression that will determine when the plastic variables are reset.

From the **Equivalent plastic strain**, **Equivalent creep strain**, or **Equivalent viscoplastic strain** list, select **Do not set** or **User defined**. The default **User defined** value is zero.

Specify the values for the components of the **Plastic strain tensor**, **Creep strain tensor**, or **Viscoplastic strain tensor**. From the list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components of the tensor.

Depending on the type of plasticity model, set additional plasticity variables.

Plasticity
- If **Armstrong–Frederick** or **Chaboche** is selected from the Kinematic Hardening list, specify the values for the components of the **Back strain tensor** ε_b. From the **Back strain tensor** list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components. The back stress tensor is then set to $\sigma_b = \frac{2}{3}C_k\varepsilon_b$.

Porous Plasticity
- From the **Porosity rate** list, select **Do not set** or **User defined**. The default **User defined** value is zero.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Plasticity** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Pressure-Dependent Plasticity** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Creep** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Viscoplasticity** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Porous Plasticity** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Soil Plasticity** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Concrete** > **Set Variables**
Solid Mechanics > **Linear Elastic Material** > **Rocks** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Plasticity** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Pressure-Dependent Plasticity** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Creep** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Viscoplasticity** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Porous Plasticity** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Soil Plasticity** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Concrete** > **Set Variables**
Solid Mechanics > **Nonlinear Elastic Material** > **Rocks** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Plasticity** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Pressure-Dependent Plasticity** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Creep** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Viscoplasticity** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Porous Plasticity** > **Set Variables**
Solid Mechanics > **Hyperelastic Material** > **Soil Plasticity** > **Set Variables**
Solid Mechanics > **Elastoplastic Soil Material** > **Set Variables**
Shell > **Linear Elastic Material, Layered** > **Plasticity** > **Set Variables**
Shell > **Hyperelastic Material, Layered** > **Plasticity** > **Set Variables**
Layered Shell > **Linear Elastic Material** > **Plasticity** > **Set Variables**
Layered Shell > **Hyperelastic Material** > **Plasticity** > **Set Variables**
Membrane > **Linear Elastic Material** > **Plasticity** > **Set Variables**
Membrane > **Linear Elastic Material** > **Creep** > **Set Variables**
Membrane > **Linear Elastic Material** > **Viscoplasticity** > **Set Variables**
Membrane > **Linear Elastic Material, Layered** > **Plasticity** > **Set Variables**
Membrane > **Nonlinear Elastic Material** > **Plasticity** > **Set Variables**
Membrane > **Nonlinear Elastic Material** > **Creep** > **Set Variables**
Membrane > **Nonlinear Elastic Material** > **Viscoplasticity** > **Set Variables**
Truss > **Linear Elastic Material** > **Plasticity** > **Set Variables**

Ribbon

Physics tab with **Plasticity**, **Pressure-Dependent Plasticity**, **Creep**, **Viscoplasticity**, **Porous Plasticity**, **Soil Plasticity**, **Concrete**, **Rocks**, or **Elastoplastic Soil Material** node selected in the model tree:

Attributes > **Set Variables**

Annealing

Use the **Annealing** subnode to Plasticity to reset plastic hardening variables (equivalent plastic strain and, in the presence of kinematic hardening, the back stress). The plastic hardening variables are reset when the temperature exceeds a specified value that you define.

The **Annealing** node requires the Nonlinear Structural Materials Module. For details, see www.comsol.com/products/specifications/

For more detailed control over the modification of plasticity variables, you can also use the Set Variables node.

ANNEALING

Enter the **Annealing temperature**. This is the temperature above which the plastic hardening variables are reset.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Plasticity** > **Annealing**

Solid Mechanics > **Nonlinear Elastic Material** > **Plasticity** > **Annealing**

Ribbon

Physics tab with **Plasticity** node selected in the model tree:

Attributes > **Annealing**

Pressure-Dependent Plasticity

Use the **Pressure-Dependent Plasticity** subnode to define the properties for modeling materials where the yielding and failure depends on pressure. This material model is

available in the Solid Mechanics interface, and can be used together with Linear Elastic Material, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module is required. For details, see www.comsol.com/products/specifications/.

When a **Pressure-Dependent Plasticity** node is present, an **Equivalent Plastic Strain** plot is available under **Result Templates**.

See also Pressure-Dependent and Soil Plasticity in the *Structural Mechanics Theory* chapter.

PLASTICITY MODEL

Use this section to define the plastic properties of the material. Select a plasticity model — **Drucker–Prager**, **Elliptic**, **Foam**, or **Parabolic**.

- The default is the **Drucker–Prager** criterion.
- Select **Elliptic** to use a general elliptic yield criterion.
- Select **Foam** to use an elliptic yield criterion specialized for crushable foam materials.
- Select **Parabolic** to use a general parabolic yield criterion.

The Drucker–Prager, Elliptic, and Parabolic options are generic plasticity models intended for pressure sensitive materials. These models share the following common settings:

The default **Initial yield stress** σ_{ys0} uses values **From material** and represents the von Mises stress where plastic deformation starts given zero pressure.

Select the **Plastic potential** Q_p related to the flow rule — **Associated** or **Nonassociated**.

Select how the **Equivalent plastic strain** ε_{pe} is computed — **Associated**, **von Mises**, or **User defined**. Enter a **User defined** expression in the h_p field as needed. See Hardening Rule for details.

Select the type of linear or nonlinear isotropic hardening model. See Isotropic Hardening for details.

Drucker–Prager

For the Drucker–Prager plasticity model, the yield function is defined by two parameters, the initial yield stress σ_{ys0} at zero pressure, and a parameter a_1 that controls the pressure dependency.

When **Plastic potential** is set to **Nonassociated**, an additional parameter a_{1Q} is used to define the pressure dependency of the plastic potential independently of the yield function.

The Drucker–Prager model can be combined with a Cap and Cutoff if needed.

See also Drucker–Prager Criterion in the *Structural Mechanics Theory* chapter.

Elliptic

The yield function of the elliptic plasticity model can be modified by specifying parameters b_1, b_2, and b_3. This set of parameters defines the shape of the ellipse in the Meridional Plane through the function

$$M_b(\sigma_m) = b_1\sigma_m + b_2^2(\sigma_m - b_3)^2$$

where σ_m is the mean stress. Each parameter can be taken **From Material**, or given as a **User defined** value. It is also possible to set a parameter to **Exclude**, in which case that term of the function is explicitly removed.

In addition, the shape of the yield function in the Octahedral Plane can be controlled by setting the **Octahedral section** $\Gamma(\theta)$ — **Circular**, **Gudehus**, or **Lagioia–Panteghini**.

- For **Gudehus**, enter the **Strength ratio** c.
- For **Lagioia–Panteghini**, enter the parameters c_1, c_2, c_3.

See Octahedral Section in the *Structural Mechanics Theory* chapter for details.

When **Plastic potential** is set to **Nonassociated**, an additional set of parameters b_{1Q}, b_{2Q}, and b_{3Q} can be specified. This set of parameters defines the shape of the plastic potential in the Meridional Plane independently from the yield function. Each parameter can either be taken **From Material**, or assigned a **User defined** value. It is also possible to set a parameter to **Exclude**, in which case that term in the nonassociated potential is explicitly removed.

The shape of the plastic potential is assumed to be circular in the octahedral plane.

See also Elliptic Criterion in the *Structural Mechanics Theory* chapter.

Parabolic

The yield function of the parabolic plasticity model can be modified by specifying the parameters a_1, a_2, a_3, and a_4. The set of parameters defines the shape of the parabola in the Meridional Plane by the function

$$M_a(\sigma_m) = a_1\sigma_m + a_2\sigma_m^2 + a_3\exp(a_4\sigma_m)$$

where σ_m is the mean stress. Each parameter can be taken **From Material**, or given a **User defined** value. It is also possible to **Exclude** a parameter, in which case that term is explicitly removed from the yield function.

In addition, the shape of the yield function in the Octahedral Plane can be controlled by setting **Octahedral section** $\Gamma(\theta)$ — **Circular**, **Gudehus**, or **Lagioia–Panteghini**. See Octahedral Section in the *Structural Mechanics Theory* chapter for details.

- For **Gudehus**, enter the **Strength ratio** c.
- For **Lagioia–Panteghini**, enter the parameters c_1, c_2, c_3.

When **Plastic potential** is set to **Nonassociated**, an additional set of parameters a_{1Q}, a_{2Q}, a_{3Q}, and a_{4Q} can be specified. This set of parameters defines the shape of the plastic potential in the meridional plane independently from the yield function. Each parameter can either be taken **From Material**, or given a **User defined** value. It is also possible to set a parameter to **Exclude**, in which case that term of the function is explicitly removed. The shape of the plastic potential is assumed to be circular in the Octahedral Plane.

The parabolic plasticity model can be combined with a Cap and Cutoff if needed.

See also Parabolic Criterion in the *Structural Mechanics Theory* chapter.

Foam

The Foam plasticity model is a specialized version of the Elliptic model often used for the analysis of structural foam and other materials with a cellular structure. The material model is set up to make it straightforward to specify the anisotropic strength

of foams when loaded in tension and compression. In addition, a volumetric hardening model can be included to account for the compaction of the foam during plastic deformation.

Define the initial shape of the elliptic yield function by specifying the following parameters:

- **Initial hydrostatic yield stress, compression** p_{c0}.
- **Hydrostatic yield stress, tension** p_t.
- **Initial uniaxial compressive yield stress** σ_{uc0}.

The default values are taken **From material**, but optionally a **User defined** input can be entered.

Foam plasticity uses a nonassociated flow rule where the plastic potential is an ellipse defined by a single parameter, the **plastic Poisson's ratio** ν_p. It is identified as the ratio of longitudinal to transverse plastic strain, that is, the value of ν_p controls to amount of volumetric plastic deformation when loading is not on the hydrostatic axis, such as in uniaxial compression. The value of ν_p can be taken **From material**, a **User defined** input, or **From default suggestion**. The default suggestion corresponds to $\nu_p = 0$, which is common for many structural foams.

The plastic Poisson's ratio ν_p must be defined between -1 and 0.5.

A **Volumetric hardening model** can be added to the foam plasticity model by entering a hardening function. When **Perfectly plastic** is selected no hardening occurs. Note that the yield stress in hydrostatic tension, p_t, is always treated as constant during plastic deformation.

By selecting **Hardening function**, it is possible to define the variation of the compressive hydrostatic yield stress by specifying a function p_{ch} with respect to the accumulated volumetric plastic strain, ε_{pv}. The variable ε_{pv} is positive in compression and it increases monotonically. The total hydrostatic yield stress is then given by

$$p_c = p_{c0} + p_{ch}(\varepsilon_{pv})$$

The function p_{ch} can either be taken **From material**, or given by a **User defined** input. In the latter case, enter an expression where the variable for ε_{pv} is named using the scheme `<physics>.epev`.

Alternatively, it is possible to specify evolution of the uniaxial compressive yield stress. When **Hardening function, uniaxial data**, is selected, define the variation of the uniaxial compressive yield stress by specifying a function σ_{uch} with respect to the accumulated axial plastic strain, ε_{pa}. The axial plastic strain is here estimated as $\varepsilon_{pa} = \varepsilon_{pv}/(1 - 2\nu_p)$, and the total uniaxial compressive yield stress is then given by

$$\sigma_{uc} = \sigma_{uc0} + \sigma_{uch}(\varepsilon_{pa})$$

The function σ_{uch} can either be taken **From material**, or given by a **User defined** input. In the latter case, enter an expression where the variable for ε_{pa} is named using the scheme `<physics>.epea`.

See also Foam Criterion in the *Structural Mechanics Theory* chapter.

To compute the energy dissipation caused by plasticity, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

NONLOCAL PLASTICITY MODEL

Nonlocal plasticity can be used to facilitate, for example, the modeling of material softening. Typical examples that involve material softening are finite-strain plasticity and soil plasticity. In these situations, standard (local) plasticity calculations reveal a mesh and topology dependence, where a mesh refinement fails to reproduce a physically sound solution. Nonlocal plasticity adds regularization to the equivalent plastic strain, thereby stabilizing the solution.

The default setting is **None**. Select **Implicit Gradient** to add nonlocal regularization to the equivalent plastic strain. Then, enter values or expressions for:

- **Length scale**, l_{int} — The regularization length scale for the equivalent plastic strain ε_{pe} should ideally be greater than the largest mesh element size.
- **Nonlocal coupling modulus**, H_{nl} — The value represents the penalization of the difference between the local and nonlocal variables. A larger value enforces the equivalent plastic strain ε_{pe} to be closer to the nonlocal equivalent plastic strain

$\varepsilon_{pe,nl}$. A typical value would be around 10% the equivalent shear modulus of the material.

See also Nonlocal Plasticity in the *Structural Mechanics Theory* chapter.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog, and it is available with the **Implicit gradient** nonlocal plasticity model.

Select the shape function for the **Nonlocal equivalent plastic strain** $\varepsilon_{pe,nl}$ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The plastic potential includes a vertex where it intersects the hydrostatic axis. To improve robustness of the plasticity algorithm, smoothing is applied to remove these singularities.

Select the **Smoothing of plastic potential** — **Automatic**, **Manual tuning**, or **User defined** to control how much the smoothed plastic potential deviates from the original potential.

When **Manual tuning** or **User defined** is selected, enter the **Edge smoothing parameter** β when using Mohr–Coulomb plastic potential. The default value is $\beta = 0.99$.

When **Manual tuning** is selected, enter a value for the **Vertex smoothing multiplier** f_v. This multiplier scales the amount of smoothing at the apex given by the **Automatic** option.

The **User defined** option allows full control of the smoothing by entering a value for the **Vertex smoothing parameter** $\sigma_{v,off}$. A typical value would be around 10% of the initial cohesion for soil models.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** or **Backward Euler, damped** is selected,

it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

When the **Backward Euler, damped** method is selected, the Newton's method is enhanced by line search iterations. Using this method can improve the robustness of the plasticity algorithm when the plastic potential or hardening model are highly nonlinear. When selected, it is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

See also Apex Smoothing, Vertex Smoothing, and Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Pressure-Dependent Plasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Pressure-Dependent Plasticity**
Solid Mechanics > **Hyperelastic Material** > **Pressure-Dependent Plasticity**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Pressure-Dependent Plasticity**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Pressure-Dependent Plasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Pressure-Dependent Plasticity**

Creep

Use the **Creep** subnode to define the creep properties of the material. This material model is available in the Solid Mechanics, Shell, Layered Shell, and Membrane

interfaces, and can be used together with Linear Elastic Material, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module or the Geomechanics Module are required, the available options depend on the used products. For details, see www.comsol.com/products/specifications/.

When a **Creep** node is present, an **Equivalent Creep Strain** plot is available under **Result Templates**.

See also Creep and Viscoplasticity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is present when **Creep** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Creep node in the Layered Shell chapter.
- **Hyperelastic Material** in the Layered Shell interface. See the documentation for the Creep node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Creep node in the Shell and Plate chapter.
- **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Creep node in the Shell and Plate chapter.

Linear Elastic Material, Layered in the Membrane interface. See the documentation for the Creep node in the Membrane chapter.

CREEP MODEL

Select a **Material model** — **Norton**, **Garofalo (hyperbolic sine)**, **Nabarro–Herring**, **Coble**, **Weertman**, or **User defined**.

Norton

For **Norton**, enter the following settings:

- **Creep rate coefficient** A.
- **Reference stress** σ_{ref}. The default is 1 MPa.

- **Stress exponent** n.

Garofalo (Hyperbolic Sine)

For **Garofalo (hyperbolic sine)**, enter the following settings:

- **Creep rate coefficient** A.
- **Reference stress** σ_{ref}. The default is 1 MPa.
- **Stress exponent** n.

Nabarro–Herring

For **Nabarro–Herring**, enter the following settings:

- **Volume diffusivity** D_{v}.
- **Burgers vector** b.
- **Grain diameter** d.

Coble

For **Coble** enter the following settings:

- **Grain boundary diffusivity** D_{gb}.
- **Burgers vector** b.
- **Grain diameter** d.

Weertman

For **Weertman**, enter the following settings:

- **Diffusivity** D.
- **Burgers vector** b.
- **Stress exponent** n.
- **Reference stress** σ_{ref}. The default is 1 MPa.

User Defined

For **User defined**, enter an expression for the creep rate f as a function of the equivalent stress σ_{e}. The default expression is `<item>.sequ/1[Pa*s]`, where `<item>` is the name of the creep node.

For **Norton**, **Garofalo (hyperbolic sine)**, **Nabarro–Herring**, **Coble**, **Weertman**, or **User defined**, select an **Equivalent stress** — **von Mises** (the default), **Hill orthotropic**, **pressure**, or **User defined**.

ISOTROPIC HARDENING MODEL

Select the isotropic hardening function h — **None**, **Strain hardening**, **Time hardening**, or **User defined**.

Strain hardening

For **Strain hardening**, enter the following settings:

- **Hardening exponent** m. The default is 0.
- **Equivalent creep strain shift** $\varepsilon_{\text{shift}}$. The default is 1e-5.
- **Reference time** t_{ref}. The default is 1 h.

Time hardening

For **Time hardening**, enter the following settings:

- **Hardening exponent** m. The default is 0.
- **Time shift** t_{shift}. The default is 0 s.
- **Reference time** t_{ref}. The default is 1 h.

User defined

For **User defined**, enter an expression for the hardening function h as a function of equivalent creep strain ε_{ce}, time t or any other variable in the model.

THERMAL EFFECTS

Select a thermal creep function — **None**, **Arrhenius**, or **User defined**.

Arrhenius

For Arrhenius, enter the following setting:

- **Reference temperature** T_{ref}. The default value, `Inf`, corresponds to omitting the term with T_{ref} in the Arrhenius expression.
- **Creep activation energy** Q. The default is 0 J/mol.

User defined

For **User defined**, enter an expression for the thermal creep function g as a function of temperature T or any other variable in the model.

TIME STEPPING

Select a **Method** — **Automatic**, **Backward Euler**, **Forward Euler**, or **Domain ODEs**.

The **Backward Euler** method is not available with the Layered Shell interface nor with the **Linear Elastic Material, Layered** material model in the Shell and Membrane interfaces.

Automatic

The **Automatic** method corresponds to the backward Euler method except for the Layered Shell interface or when **Linear Elastic Material, Layered** is used. Domain ODEs are solved in these cases.

Backward Euler

For the **Backward Euler** method specify:

- **Maximum number of local iterations**. To define the maximum number of iterations of the Newton loop when solving the local creep equations.
- **Absolute tolerance**. To define the absolute tolerance for convergence of the local creep equations. Convergence is judged based on the step size, that is, the size of the Newton correction, of each equation.
- **Relative tolerance**. To define the relative tolerance for convergence of the local creep equations. Convergence is judged based on the step size, that is, the size of the Newton correction, of each equation.
- **Residual tolerance**. To define the tolerance for convergence of the local creep equations based on the residual of each equation.

Setting either the **Absolute tolerance** and **Relative tolerance**, or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all settings are set to zero. If both a step size and residual convergence check is requested, it is sufficient that one of the conditions is satisfied for convergence.

Forward Euler

For the **Forward Euler** method, enter the **Maximum creep strain increment**.

It is recommend to reset the solver settings to its default values when selecting the forward Euler method, since the method is only conditionally stable. This will add a **Maximum step constraint** to the **Time-Dependent Solver** based on an estimate of the stability limit used to update the creep equations. The value entered in the **Maximum creep strain increment** field is taken in to account, which can be used to improve the accuracy of the method. If the solver sequence cannot be reset, the stability limit can be entered manually in the **Time-Dependent Solver** settings by using the variable `<item>.tmax`, where `<item>` is the name of the creep node.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of these fields can affect the convergence of the overall solution.

ADVANCED

For a 2D geometry, enable **Include out-of-plane strains** to solve for all components of the creep strain tensor. By default, COMSOL Multiphysics assumes that the out-of-plane components are zero.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

To compute the energy dissipation caused by creep, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Creep**
Solid Mechanics > **Nonlinear Elastic Material** > **Creep**
Solid Mechanics > **Hyperelastic Material** > **Creep**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Creep**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Creep**
Shell > **Linear Elastic Material, Layered** > **Creep**
Shell > **Hyperelastic Material, Layered** > **Creep**
Layered Shell > **Linear Elastic Material** > **Creep**
Layered Shell > **Hyperelastic Material** > **Creep**
Membrane > **Linear Elastic Material** > **Creep**
Membrane > **Linear Elastic Material, Layered** > **Creep**
Membrane > **Nonlinear Elastic Material** > **Creep**
Membrane > **Hyperelastic Material** > **Creep**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

Attributes > **Creep**

Additional Creep

Use the **Additional Creep** subnode to define additional contributions to the parent Creep node, such as primary or tertiary creep behavior. A **Creep** node can have any number of **Additional Creep** subnodes with different settings to model advanced creep

mechanisms. Note that the time-integrated creep strain ε_{cr} is computed for the sum of all creep rate contributions; the same applies for the equivalent creep strain ε_{ce}.

The Nonlinear Structural Materials Module or the Geomechanics Module are required for this feature, and the available options depend on the used products. For details, see www.comsol.com/products/specifications/.

See also Creep and Viscoplasticity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is present when the parent **Creep** node is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Additional Creep node in the Layered Shell chapter.
- **Hyperelastic Material** in the Layered Shell interface. See the documentation for the Additional Creep node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Additional Creep node in the Shell and Plate chapter.
- **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Additional Creep node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Additional Creep node in the Membrane chapter.

CREEP MODEL

The available settings in this section are the same as for the Creep node.

The **Equivalent stress** setting has an additional option — **From parent**, which is the default. When selected, the Additional Creep subnode will use the same equivalent stress σ_e as in the parent Creep node.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Creep** > **Additional Creep**
Solid Mechanics > **Nonlinear Elastic Material** > **Creep** > **Additional Creep**
Solid Mechanics > **Hyperelastic Material** > **Creep** > **Additional Creep**
Shell > **Linear Elastic Material, Layered** > **Creep** > **Additional Creep**
Shell > **Hyperelastic Material, Layered** > **Creep** > **Additional Creep**
Layered Shell > **Linear Elastic Material** > **Creep** > **Additional Creep**
Layered Shell > **Hyperelastic Material** > **Creep** > **Additional Creep**
Membrane > **Linear Elastic Material** > **Creep** > **Additional Creep**
Membrane > **Linear Elastic Material, Layered** > **Creep** > **Additional Creep**
Membrane > **Nonlinear Elastic Material** > **Creep** > **Additional Creep**
Membrane > **Hyperelastic Material** > **Creep** > **Additional Creep**

Ribbon

Physics tab with **Creep** node selected in the model tree:

Attributes > **Additional Creep**

Viscoplasticity

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model. This material is available in the Solid Mechanics, Layered Shell, Shell, and Membrane interfaces, and can be used together with Linear Elastic Material, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

When a **Viscoplasticity** node is present, an **Equivalent Viscoplastic Strain** plot is available under **Result Templates**.

See also Creep and Viscoplasticity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is present when **Viscoplasticity** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Viscoplasticity node in the Layered Shell chapter.
- **Hyperelastic Material** in the Layered Shell interface. See the documentation for the Viscoplasticity node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Viscoplasticity node in the Shell and Plate chapter.
- **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Viscoplasticity node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Viscoplasticity node in the Membrane chapter.

VISCOPLASTICITY MODEL

Select a **Material model** — **Anand**, **Anand–Narayan**, **Bingham**, **Chaboche**, **Peric**, **Perzyna**, or **User defined**.

Anand

For **Anand** enter the following data:

- **Viscoplastic rate coefficient** A.
- **Stress multiplier** ξ.
- **Stress sensitivity** m.
- **Deformation resistance saturation coefficient** s_{sat}.
- **Deformation resistance initial value** s_{init}.
- **Hardening coefficient** h_0.
- **Hardening sensitivity** a.
- **Deformation resistance sensitivity** n.

Each of the material properties can either be defined obtained **From material**, or as **User defined**. In the latter case, enter a value or an expression.

Viscoplastic Creep in Solder Joints: Application Library path **Nonlinear_Structural_Materials_Module/Viscoplasticity/viscoplastic_solder_joints**

Anand–Narayan

For **Anand–Narayan** enter the following data:

- **Viscoplastic rate coefficient** A.
- **Stress multiplier** ξ.
- **Stress sensitivity** m.
- **Deformation resistance saturation coefficient** s_{sat}.
- **Deformation resistance initial value** s_{init}.
- **Hardening coefficient** h_0.
- **Hardening sensitivity** a.
- **Deformation resistance sensitivity** n.

Each of the material properties can either be defined obtained **From material**, or as **User defined**. In the latter case, enter a value or an expression.

Bingham

Enter the **Viscosity** η. The default is 1 Pa s.

Chaboche

For **Chaboche** enter the following settings:

- **Viscoplastic rate coefficient** A.
- **Reference stress** σ_{ref}.
- **Stress exponent** n.

Peric

For **Peric** enter the following settings:

- **Viscoplastic rate coefficient** A.
- **Stress exponent** n.

Perzyna

For **Perzyna** enter the following settings:

- **Viscoplastic rate coefficient** A.
- **Reference stress** σ_{ref}. The default is 1 MPa.

User Defined

For **User defined** enter a value or an expression for the **Equivalent viscoplastic rate**. Write any expression in terms of stress tensor components or its invariants in the λ_{vp} field.

Plasticity Model

For **Bingham**, **Chaboche**, **Peric**, **Perzyna**, or **User defined** viscoplasticity models, define the Plasticity Model, the Isotropic Hardening, and the Kinematic Hardening as needed.

The yield function defines the limit of the elastic regime, $F(\sigma_e, \sigma_{ys}) \leq 0$, and sets the onset for viscoplastic deformation. Changing the equivalent stress measure allows to specify different criteria. See Yield Criterion for details.

Select the **Equivalent stress** — **von Mises**, **Tresca**, **Hill orthotropic**, or **User defined** to define the yield criterion.

- The default is **von Mises** criterion with associate plastic potential.
- Select **Tresca** to use Tresca yield criterion. The plastic potential can be an **Associated** or nonassociated flow rule with the **von Mises** stress as plastic potential.
- Select **Hill orthotropic** to use Hill's criterion. From the **Specify** list select either the **Initial tensile and shear yield stresses** σ_{ys0ij} or **Hill's coefficients** F, G, H, L, M, and N. The default for either selection uses values **From material** (if it exists) or **User defined**. The principal directions of orthotropy are inherited from the coordinate system selection in the parent node. See Expressions for Hill's Coefficients for details.
- For **User defined** enter a value or an expression for the equivalent stress. Write any expression in terms of stress tensor components or its invariants in the σ_e field.

For all viscoplastic models, the default **Initial yield stress** σ_{ys0} uses values **From material** and represents the stress level where viscoplastic deformation starts.

Select the **Plastic potential** Q_{vp} related to the flow rule — **Associated**, **von Mises**, or **User defined** (nonassociated). Enter a **User defined** value in the Q_{vp} field as needed.

Isotropic Hardening

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list. See Numerical Integration Algorithm for details.

- Select **Perfectly plastic** (ideal viscoplasticity) if the material can undergo viscoplastic deformation without any increase in yield stress.
- For **Linear** the default **Isotropic tangent modulus** $E_{T\mathrm{iso}}$ uses values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified as hardening occurs, and it is related to the equivalent viscoplastic strain $\varepsilon_{\mathrm{vpe}}$ as

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + E_{\mathrm{iso}}\varepsilon_{\mathrm{vpe}} \text{ with } \frac{1}{E_{\mathrm{iso}}} = \frac{1}{E_{T\mathrm{iso}}} - \frac{1}{E}$$

  For the linear isotropic hardening model, the yield stress increases proportionally to the equivalent viscoplastic strain $\varepsilon_{\mathrm{vpe}}$. The Young's modulus E is taken from the elastic material properties.
- Select **Ludwik** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the power law

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + k\varepsilon_{\mathrm{vpe}}^{n}$$

  the **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.
- For **Swift** nonlinear isotropic hardening, the **Reference strain** ε_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified by the power law

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}}\left(1 + \frac{\varepsilon_{\mathrm{vpe}}}{\varepsilon_0}\right)^{n}$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{sat}}\left(1 - e^{-\beta\varepsilon_{\mathrm{vpe}}}\right)$$

  the **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.
- For **Hockett–Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_{∞}, the **Saturation coefficient** m, and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{ys} = \sigma_{ys0} + (\sigma_\infty - \sigma_{ys0})\left(1 - e^{-m\varepsilon_{vpe}^n}\right)$$

- For **Hardening function**, the isotropic **Hardening function** $\sigma_h(\varepsilon_{vpe})$ uses values **From material** or **User defined**. The yield level σ_{ys} is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\varepsilon_{vpe})$$

  This definition implies that the hardening function $\sigma_h(\varepsilon_{vpe})$ in the Material node must be zero at zero viscoplastic strain. In other words, $\sigma_{ys} = \sigma_{ys0}$ when $\varepsilon_{vpe} = 0$. With this option it is possible to enter any nonlinear isotropic hardening curve. The hardening function can depend on more variables than the equivalent viscoplastic strain, for example the temperature. Select **User defined** to enter any function of the equivalent viscoplastic strain ε_{vpe}. The variable is named using the scheme `<physics>.evpe`, for example, `solid.evpe`.

Kinematic Hardening

Select the type of kinematic hardening model from the **Kinematic hardening model** list. See Kinematic Hardening for details.

- Select **No kinematic hardening** when the material can undergo viscoplastic deformation without a shift in the yield surface.
- If **Linear** is selected as the **Kinematic hardening model**, the default **Kinematic tangent modulus** E_k uses values **From material**. This parameter is used to calculate the back stress σ_b as

$$\sigma_b = \frac{2}{3}C_k \cdot \varepsilon_{vp} \text{ with } \frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

  This is Prager's linear kinematic hardening model, so the back stress σ_b is collinear to the viscoplastic strain tensor ε_{vp}.

- If **Armstrong–Frederick** is selected from the list, the default **Kinematic hardening modulus** C_k and **Kinematic hardening parameter** γ_k use values **From material**. These parameters are used to calculate the back stress σ_b from the rate equation

$$\dot{\sigma}_b = \frac{2}{3}C_k\dot{\varepsilon}_{vp} - \gamma_k\dot{\varepsilon}_{vpe}\sigma_b$$

  This is the Armstrong–Frederick nonlinear kinematic hardening model.

- When **Chaboche** is selected from the **Kinematic hardening model** list, the default **Kinematic hardening modulus** C_0 uses values **From material**. Add branches as needed to solve N rate equations for the back stresses:

$$\dot{\sigma}_{b,i} = \frac{2}{3} C_i \dot{\varepsilon}_{vp} - \gamma_i \dot{\varepsilon}_{vpe} \sigma_{b,i}$$

For each **Branch** row, enter C_i (the hardening modulus of the branch i) in the **Hardening modulus (Pa)** column and γ_i (the hardening parameter of the branch i) in the **Hardening parameter (1)** column.

Use the **Add** button (+) and the **Delete** button ( ) to add or delete a row in the table. Use the **Load from file** button ( ) and the **Save to file** button ( ) to load and store data for the branches in a text file with three space-separated columns (from left to right): the branch number, the hardening modulus for that branch, and the hardening parameter for that branch.

The total back stress σ_b is then computed from the sum

$$\sigma_b = \frac{2}{3} C_0 \varepsilon_{vp} + \sum_{i=1}^{N} \sigma_{b,i}$$

THERMAL EFFECTS

Select a thermal function $g(T)$ — **None**, **Arrhenius**, or **User defined**, which acts as a multiplier for the viscoplastic rate.

Arrhenius

For **Arrhenius**, enter the following setting:

- **Reference temperature** T_{ref}. The default value is 293.15 K.
- **Activation energy** Q. The default is 0 J/mol.

User defined

For **User defined**, enter an expression for $g(T)$ as a function of temperature T or other variables in the model.

This section is not present when **Viscoplasticity** is used with:

- The Layered Shell interface.
- **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** in the Shell interface.
- **Linear Elastic Material, Layered** in the Membrane interface.

In the above cases, the Domain ODEs method is always used.

Select a **Method** — **Automatic**, **Backward Euler**, or **Domain ODEs**.

Automatic

The **Automatic** method corresponds to the backward Euler method with predefined settings for the Newton loop used to solve the local equations.

Backward Euler

For the **Backward Euler** method, enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local viscoplasticity equations.
- **Absolute tolerance**. To check the convergence of the local viscoplasticity equations based on the step size in the Newton loop.
- **Relative tolerance**. To check the convergence of the local viscoplasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value.
- **Residual tolerance**. To check the convergence of the local viscoplasticity equations based on the residual of each equation.

If both a step size and residual convergence check is requested, it is sufficient that either of the conditions are fulfilled. Setting either the **Absolute tolerance** and **Relative tolerance** or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all settings are set to zero.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of these fields can affect the convergence of the overall solution.

ADVANCED

For a 2D geometry, enable **Include out-of-plane strains** to solve for all components of the viscoplastic strain tensor. By default, COMSOL assumes that the out-of-plane components are zero.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

To compute the energy dissipation caused by viscoplasticity, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Viscoplasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Viscoplasticity**
Solid Mechanics > **Hyperelastic Material** > **Viscoplasticity**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Viscoplasticity**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Viscoplasticity**
Shell > **Linear Elastic Material, Layered** > **Viscoplasticity**
Shell > **Hyperelastic Material, Layered** > **Viscoplasticity**
Layered Shell > **Linear Elastic Material** > **Viscoplasticity**
Layered Shell > **Hyperelastic Material** > **Viscoplasticity**
Membrane > **Linear Elastic Material** > **Viscoplasticity**
Membrane > **Linear Elastic Material, Layered** > **Viscoplasticity**
Membrane > **Nonlinear Elastic Material** > **Viscoplasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material**, **Hyperelastic Material, Layered**, or **Nonlinear Elastic Material** node selected in the model tree:

Attributes > **Viscoplasticity**

Polymer Viscoplasticity

Use the **Polymer Viscoplasticity** subnode to define the viscoplastic properties of rubber, elastomer, and thermoplastic materials. This material is available in the Solid

Mechanics, Layered Shell, Shell, and Membrane interfaces, and can be used together with Hyperelastic Material.

See also Polymer Viscoplasticity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

This section is present when **Polymer Viscoplasticity** is used as a subnode to:

- **Hyperelastic Material** in the Layered Shell interface. See the documentation for the Polymer Viscoplasticity node in the Layered Shell chapter.
- **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Polymer Viscoplasticity node in the Shell and Plate chapter.

VISCOPLASTICITY MODEL

Use this section to define the viscoplastic properties of the material.

Select a **Material model** — **Bergstrom–Boyce**, **Bergstrom–Bischoff**, **Parallel network**, or **User defined**. Then follow the instructions below.

Bergstrom–Boyce

The rheology of the Bergstrom–Boyce model adds a spring-dashpot network in parallel to the **Hyperelastic Material**.

Enter the **Energy factor** β_v to describe the stiffness of the spring element in the network. The default value is 1.2, which means that at infinitesimal deformations the shear modulus of the network is 1.2 times the equivalent shear modulus of the **Hyperelastic Material**.

Enter the settings for the nonlinear dashpot:

- **Viscoplastic rate coefficient** A.
- **Flow resistance** σ_{res}.
- **Stress exponent** n.
- **Cutoff stress** σ_{co}.

Select the hardening model from the **Isotropic hardening model** list.

- Select **None** if the material undergoes viscoplastic deformation without hardening.
- For **Strain hardening** enter a value or an expression for the strain exponent c.
- For **User defined** enter a value or an expression for the hardening function. Write any expression in terms of the viscoplastic strain in the $h(\lambda_{\mathrm{vpe}})$ field.

Select the **Stiffness used in stationary studies** — **Long-term** or **Instantaneous**. With **Long-term** the damper is assumed to be fully relaxed, whereas with **Instantaneous** the damper is assumed to be rigid.

Bergstrom–Bischoff

The rheology of the Bergstrom–Bischoff model consists of two spring-dashpot networks added in parallel to a **Hyperelastic Material**.

Enter the **Viscoplastic rate coefficient** A.

Enter the settings for the nonlinear dashpot in the first network:

- **Energy factor** $\beta_{\mathrm{v}1}$.
- **Flow resistance** $\sigma_{1,\mathrm{res}}$.
- **Stress exponent** n_1.
- **Pressure hardening coefficient** a_1.

Enter the settings for the nonlinear dashpot in the second network:

- **Initial energy factor** $\beta_{\mathrm{v,i}}$.
- **Final energy factor** $\beta_{\mathrm{v,f}}$.
- **Energy factor evolution coefficient** α.
- **Flow resistance** $\sigma_{2,\mathrm{res}}$.
- **Stress exponent** n_2.
- **Pressure hardening coefficient** a_2.

Select the **Stiffness used in stationary studies** — **Long-term** or **Instantaneous**. With **Long-term** all dampers are assumed to be fully relaxed, whereas with **Instantaneous** all dampers are assumed to be rigid.

Parallel Network

The rheology of the *Parallel Network* model consists of one or more spring-dashpot networks added in parallel to a **Hyperelastic Material**.

Select a **Material model** for the spring — **From parent**, **Neo-Hookean**, **Mooney–Rivlin**, **Yeoh**, **Ogden**, **Arruda–Boyce**, or **Gent**.

For **From parent**, enter the **Energy factor** β_v to describe the stiffness of the spring element in the network. The default value is 1.2, which means that at infinitesimal deformations the shear modulus of the network is 1.2 times the equivalent shear modulus of the parent **Hyperelastic Material**.

For the other options, enter the settings as described in Neo-Hookean, Mooney–Rivlin, Two Parameters, Yeoh, Ogden, Arruda–Boyce, or Gent.

Enter the settings for the nonlinear dashpot:

- **Viscoplastic rate coefficient** A.
- **Flow resistance** σ_{res}.
- **Stress exponent** n.
- **Cutoff stress** σ_{co}.
- **Pressure hardening coefficient** a.

Select the hardening model from the **Isotropic hardening model** list.

- Select **None** if the material undergoes viscoplastic deformation without hardening.
- For **Strain hardening** (default) enter a value or an expression for the strain exponent c.
- For **User defined** enter a value or an expression for the hardening function. Write any expression in terms of the viscoplastic strain in the $h(\lambda_{vpe})$ field.

Select the **Stiffness used in stationary studies** — **Long-term** or **Instantaneous**. With **Long-term** all dampers are assumed to be fully relaxed, whereas with **Instantaneous** all dampers are assumed to be rigid.

Additional Network

When the Parallel Network model is selected, it is possible to add **Additional Network** subnodes to the **Polymer Viscoplasticity** node.

Enter the **Energy factor** β_v to describe the stiffness of the spring element in the additional network. The default value is 1.2, which means that at infinitesimal deformations the shear modulus of the network is 1.2 times the equivalent shear modulus of the parent Parallel Network.

The settings for the additional nonlinear dashpot are the same as described for the Parallel Network model.

User Defined

The rheology of the *User defined* model consists of one spring-dashpot network added in parallel to a **Hyperelastic Material**.

Enter the **Energy factor** β_v to describe the stiffness of the spring element in the network. The default value is 1.2, which means that at infinitesimal deformations the shear modulus of the network is 1.2 times the equivalent shear modulus of the parent **Hyperelastic Material**.

For **User defined** enter a value or an expression for the **Equivalent viscoplastic rate**. Write any expression in terms of stress tensor components or its invariants in the λ_{vp} field.

Select the **Stiffness used in stationary studies** — **Long-term** or **Instantaneous**. With **Long-term** the damper is assumed to be fully relaxed, whereas with **Instantaneous** the damper is assumed to be rigid.

THERMAL EFFECTS

Select a thermal function $g(T)$ — **None**, **Arrhenius**, **Power law**, or **User defined**, which acts as a multiplier for the viscoplastic rate.

Arrhenius

For **Arrhenius**, enter the following setting:

- **Reference temperature** T_{ref}. The default value is 293.15 K.
- **Activation energy** Q. The default is 0 J/mol.

Power law

For **Power law**, enter the following setting:

- **Reference temperature** T_{ref}. The default value is 293.15 K.
- **Temperature exponent** m. The default is 0.

User defined

For **User defined**, enter an expression for $g(T)$ as a function of temperature T or other variables in the model.

TIME STEPPING

Select a **Method** — **Automatic**, **Backward Euler**, or **Domain ODEs**.

Automatic

The **Automatic** method corresponds to the backward Euler method with predefined settings for the Newton loop used to solve the local equations.

Backward Euler

For the **Backward Euler** method, enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local viscoplasticity equations.
- **Absolute tolerance**. To check the convergence of the local viscoplasticity equations based on the step size in the Newton loop.
- **Relative tolerance**. To check the convergence of the local viscoplasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value.
- **Residual tolerance**. To check the convergence of the local viscoplasticity equations based on the residual of each equation.

If both a step size and residual convergence check is requested, it is sufficient that either of the conditions are fulfilled. Setting either the **Absolute tolerance** and **Relative tolerance** or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all are set to zero.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of these fields can affect the convergence of the overall solution.

ADVANCED

For a 2D geometry, enable **Include out-of-plane strains** to solve for all components of the viscoplastic strain tensor. By default, COMSOL assumes that the out-of-plane components are zero.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

To compute the energy dissipation caused by viscoplasticity, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent **Hyperelastic Material** node.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Hyperelastic Material** > **Polymer Viscoplasticity**
Solid Mechanics > **Thin Layer** > **Hyperelastic Material** > **Polymer Viscoplasticity**

Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Polymer Viscoplasticity**
Shell> **Hyperelastic Material, Layered** > **Polymer Viscoplasticity**
Layered Shell > **Hyperelastic Material** > **Polymer Viscoplasticity**
Membrane> **Hyperelastic Material** > **Polymer Viscoplasticity**

Ribbon
Physics tab with **Hyperelastic Material** node selected in the model tree:

Attributes > **Polymer Viscoplasticity**

Porous Plasticity

Use the **Porous Plasticity** subnode to define the properties of a plasticity model for a porous material.

The **Porous Plasticity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). The material model is available for 3D, 2D, and 2D axisymmetry.

When a **Porous Plasticity** node is present, a **Volumetric Plastic Strain** plot and a **Current Void Volume Fraction** plot are available under **Result Templates**.

POROUS PLASTICITY MODEL

Use this section to define the plastic properties of the porous material.

Material Model
Select the **Material model** for the porous plasticity criterion — **Shima–Oyane**, **Gurson**, **Gurson–Tvergaard–Needleman**, **Fleck–Kuhn–McMeeking**, **FKM–GTN**, or **Capped Drucker–Prager**.

For **FKM–GTN**, select the **Plastic potential** — **Associated**, **Gurson–Tvergaard–Needleman**, or **Fleck–Kuhn–McMeeking**.

Shima–Oyane
For **Shima–Oyane** enter the following data:

- **Initial yield stress** σ_{ys0}.
- **Shima–Oyane parameter** α.
- **Shima–Oyane parameter** γ.

- **Shima–Oyane parameter** m.
- **Initial void volume fraction** f_0.

Gurson

For **Gurson** enter the following data:

- **Initial yield stress** σ_{ys0}.
- **Initial void volume fraction** f_0.

Gurson–Tvergaard–Needleman

For **Gurson–Tvergaard–Needleman** enter the following data:

- **Initial yield stress** σ_{ys0}.
- **Tvergaard correction coefficient** q_1.
- **Tvergaard correction coefficient** q_2.
- **Tvergaard correction coefficient** q_3.
- **Initial void volume fraction** f_0.
- Select an **Effective void volume fraction** — **Bilinear**, **Asymptotic**, or **User defined**.

For **Bilinear** and **Asymptotic**, enter the following data:

- **Critical void volume fraction** f_c.
- **Failure void volume fraction** f_f.

For **User defined**, enter the effective void volume fraction as a function of for example the void volume fraction, variable `<phys>.fcv`.

Fleck–Kuhn–McMeeking

For **Fleck–Kuhn–McMeeking** enter the following data:

- **Initial yield stress** σ_{ys0}.
- **Initial void volume fraction** f_0.
- **Maximum void volume fraction** f_{max}.

FKM–GTN

For **FKM–GTN** enter the following data:

- **Initial yield stress** σ_{ys0}.
- **Tvergaard correction coefficient** q_1.
- **Tvergaard correction coefficient** q_2.
- **Tvergaard correction coefficient** q_3.

- **Initial void volume fraction** f_0.
- **Maximum void volume fraction** $f_{\max}$.
- **Void volume fraction cutoff for GTN model** f_{GTN}.
- **Void volume fraction cutoff for FKM model** f_{FKM}.

Capped Drucker–Prager

For **Capped Drucker–Prager** enter the following data:

- Select the **Plastic potential** Q_{p} related to the flow rule — **Associated** or **Nonassociated**.
- Select how the **Equivalent plastic strain** $\varepsilon_{\mathrm{pe}}$ is computed — **Associated**, **von Mises**, or **User defined**. Enter a **User defined** expression in the h_{p} field as needed. See Hardening Rule for details.
- **Initial yield stress** σ_{ys0}.
- **Yield function parameter** a_1.
- When **Plastic potential** is set to **Nonassociated**, the **Potential parameter** $a_{1\mathrm{Q}}$.
- **Initial void volume fraction** f_0.

The material properties use values **From material** (default) or **User defined**.

Enter the data for the compression cap in the Cap Model section.

See also Drucker–Prager Criterion and Porous Plasticity in the *Structural Mechanics Theory* chapter.

Octahedral Section

The shape of the yield function in the Octahedral Plane can be controlled by setting the **Octahedral section** $\Gamma(\theta)$ — **Circular**, **Gudehus**, or **Lagioia–Panteghini**.

- For **Gudehus**, enter the **Strength ratio** c.
- For **Lagioia–Panteghini**, enter the parameters c_1, c_2, c_3.

The octahedral section for the Capped Drucker–Prager model is always circular.

See also Void Nucleation and Growth in the *Structural Mechanics Theory* chapter.

Void Growth

It is possible to **Include void nucleation in tension** or **Include void growth in shear** by selecting the corresponding checkbox.

When **Include void nucleation in tension** is selected, enter the **Void volume fraction of nucleating voids**, the **Standard deviation for void nucleation**, and the **Mean strain for void nucleation**. For each property use the value **From material** or enter a **User defined** value or expression.

When **Include void growth in shear** is selected, enter the **Void growth rate parameter**. Use the value **From material** or enter a **User defined** value or expression.

See also Void Nucleation and Growth in the *Structural Mechanics Theory* chapter.

Isotropic Hardening

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list. See Isotropic Hardening for details.

- Select **Perfectly plastic** (ideal plasticity) if the material can undergo plastic deformation without any increase in yield stress.
- For **Linear** the default **Isotropic tangent modulus** $E_{T\mathrm{iso}}$ uses values **From material** (if it exists) or **User defined**. The flow stress (yield level) σ_{ys} is modified as hardening occurs, and it is related to the equivalent plastic strain in the porous matrix $\varepsilon_{\mathrm{pe}}$ as

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + E_{\mathrm{iso}}\varepsilon_{\mathrm{pe}} \quad \text{with} \quad \frac{1}{E_{\mathrm{iso}}} = \frac{1}{E_{T\mathrm{iso}}} - \frac{1}{E}$$

  For the linear isotropic hardening model, the flow stress (yield stress) increases proportionally to the equivalent plastic strain in the porous matrix $\varepsilon_{\mathrm{pe}}$. The Young's modulus E is taken from the elastic material properties.
- Select **Ludwik** from the list to model nonlinear isotropic hardening. The flow stress (yield level) σ_{fm} is modified by the power law

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + k\varepsilon_{\mathrm{pe}}^{n}$$

  The **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.
- Select **Johnson–Cook** or **Modified Johnson–Cook** from the list to model strain rate dependent hardening. The **Strength coefficient** k, **Hardening exponent** n, **Reference**

strain rate $\dot{\varepsilon}_0$, and **Strain rate strength coefficient** C use values **From material** (if it exists) or **User defined**.

Select a **Thermal softening model** — **No thermal softening**, **Power law**, or **User defined**.

- For **Power law**, enter the **Reference temperature** T_{ref}, the **Melting temperature** T_{m}, and the **Temperature exponent**, m.
- For **User defined**, enter the **Thermal Softening function** $f(T_{\text{h}})$, the **Reference temperature** T_{ref}, and the **Melting temperature** T_{m}. The softening function $f(T_{\text{h}})$ typically depends on the built-in variable for the normalized homologous temperature T_{h} and have the properties $f(0) = 0$ and $f(1) = 1$. The variable is named using the scheme `<physics>.<elasticTag>.<plasticTag>.Th`, for example, `solid.lemm1.popl1.Th`.

The yield stress and hardening function for the Johnson–Cook model is given by

$$\sigma_{\text{ys}}(\varepsilon_{\text{pe}}) = (\sigma_{\text{ys0}} + k\varepsilon_{\text{pe}}^{n})\left(1 + C\log\left(\frac{\dot{\varepsilon}_{\text{pe}}}{\dot{\varepsilon}_0}\right)\right)(1 - f(T_{\text{h}}))$$

$$T_{\text{h}} = \frac{T - T_{\text{ref}}}{T_{\text{m}} - T_{\text{ref}}}$$

For the modified Johnson–Cook model, the yield stress is instead given by

$$\sigma_{\text{ys}}(\varepsilon_{\text{pe}}) = (\sigma_{\text{ys0}} + k\varepsilon_{\text{pe}}^{n})\left(1 + C\log\left(1 + \frac{\dot{\varepsilon}_{\text{pe}}}{\dot{\varepsilon}_0}\right)\right)(1 - f(T_{\text{h}}))$$

In the case of power law softening, $f(T_{\text{h}}) = T_{\text{h}}^{m}$.

- For **Swift** nonlinear isotropic hardening, the **Reference strain** ε_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield stress depends on the power law

$$\sigma_{\text{ys}}(\varepsilon_{\text{pe}}) = \sigma_{\text{ys0}}\left(1 + \frac{\varepsilon_{\text{pe}}}{\varepsilon_0}\right)^{n}$$

Typically, the reference strain equals the onset of plasticity, $\varepsilon_0 = \sigma_{\text{ys0}}/E$.

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level is modified by the exponential law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{pe}}\right)$$

The **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.

- For **Hockett–Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_∞, the **Saturation coefficient** m, and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level is increased by the exponential law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + (\sigma_\infty - \sigma_{ys0})\left(1 - e^{-m\varepsilon_{pe}^n}\right)$$

- For **Hardening function**, the isotropic **Hardening function** $\sigma_h(\varepsilon_{pe})$ uses values **From material** or **User defined**. The flow stress (yield level) is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\varepsilon_{pe})$$

  - This definition implies that the hardening function $\sigma_h(\varepsilon_{pe})$ in the Material node must be zero at zero plastic strain. In other words, $\sigma_{ys} = \sigma_{ys0}$ when $\varepsilon_{pe} = 0$. With this option it is possible to enter any nonlinear isotropic hardening curve. The hardening function can depend on more variables than the equivalent plastic strain in the porous matrix, for example the temperature. Select **User defined** to enter any function of the equivalent plastic strain ε_{pe}. The variable is named using the scheme `<physics>.<elasticTag>.<plasticTag>.epe`, for example, `solid.lemm1.popl1.epe`.

See also Isotropic Hardening in the *Structural Mechanics Theory* chapter.

Cap Model

Use this section the enter data for the compression cap when the Capped Drucker–Prager material model is selected.

Select the **Cap model** — **Elliptic**, **Planar**, or **Cam-clay**.

- For **Elliptic**, enter the **Initial pressure limit** p_{c0} and the **Initial ellipse centroid** p_{cc0}. The ellipse aspect ratio is then derived from the intersection of the ellipse with Drucker–Prager criterion.
- For **Planar**, enter the **Initial pressure limit** p_{c0}.
- For **Cam-clay**, enter the **Initial pressure limit** p_{c0}.

Select the **Hardening model** — **Perfectly plastic** (no hardening), **Linear**, **Exponential**, or **Hardening function**.

- When **Linear** is selected from the list, the default **Hardening modulus** K_c is taken **From material**.
- When **Exponential** is selected from the list, the default **Hardening modulus** K_c and the **Maximum volumetric plastic strain** $\varepsilon_{pvol,max}$ are taken **From material**.
- When **Hardening function** is selected from the list, the default **Hardening modulus** p_{ch} and the **Maximum volumetric plastic strain** $\varepsilon_{pvol,max}$ are taken **From material**.
- Select the **Hardening variable** — **Volumetric plastic strain**, **Equivalent plastic strain**, or **User defined**.

See also Compression Cap in the *Structural Mechanics Theory* chapter.

NONLOCAL PLASTICITY MODEL

Nonlocal plasticity can be used to facilitate, for example, the modeling of material softening. Typical examples that involve material softening are finite-strain plasticity and soil plasticity. In these situations, standard (local) plasticity calculations reveal a mesh and topology dependence, where a mesh refinement fails to reproduce a physically sound solution. Nonlocal plasticity adds regularization to the equivalent plastic strain, thereby stabilizing the solution.

The default setting is **None**. Select **Implicit Gradient** to add nonlocal regularization to the equivalent plastic strain. Then, enter values or expressions for:

- **Length scale**, l_{int} — The regularization length scale for the equivalent plastic strain ε_{pe} should ideally be greater than the largest mesh element size.
- **Nonlocal coupling modulus**, H_{nl} — The value represents the penalization of the difference between the local and nonlocal variables. A larger value enforces the equivalent plastic strain ε_{pe} to be closer to the nonlocal equivalent plastic strain $\varepsilon_{pe,nl}$. A typical value would be around 10% the equivalent shear modulus of the material.

See also Nonlocal Plasticity in the *Structural Mechanics Theory* chapter.

It is possible to apply the implicit gradient method to the variables for the equivalent plastic strain or void volume fraction, by selecting the corresponding checkbox.

Select the **Void volume fraction** checkbox in order to apply the implicit gradient method of regularization to the void volume fraction variable. Enter a value for the **Length scale, void volume fraction**, $l_{\mathrm{int,f}}$.

The length scale for the equivalent plastic strain, l_{int}, and the length scale for the void volume fraction, $l_{\mathrm{int,f}}$, can be chosen independently. Ideally, both parameters should be larger than the largest mesh element size, while remaining independent of it.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog, and it is available with the **Implicit gradient** nonlocal plasticity model.

Select the shape function for the **Nonlocal equivalent plastic strain** $\varepsilon_{\mathrm{pe,nl}}$ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

When the **Void volume fraction** checkbox is selected, select the shape function for the **Nonlocal void volume fraction** f_{nl} — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Enter the **Maximum damage**, which defines the residual stiffness of the model. The default value is 0.995.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** or **Backward Euler, damped** is selected,

it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

When the **Backward Euler, damped** method is selected, the Newton's method is enhanced by line search iterations. Using this method can improve the robustness of the plasticity algorithm when the plastic potential or hardening model are highly nonlinear. When selected, it is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

For the **Capped Drucker–Prager** model, see the settings in the **Advanced** section for Pressure-Dependent Plasticity and Cap and Cutoff.

See also the Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

To compute the energy dissipation caused by porous compaction, enable the **Calculate dissipated energy** checkbox in the Energy Dissipation section of the parent material node.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Porous Plasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Porous Plasticity**
Solid Mechanics > **Hyperelastic Material** > **Porous Plasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Porous Plasticity**

Soil Plasticity

In the **Soil Plasticity** subnode you define the properties for modeling materials exhibiting soil behavior. **Soil Plasticity** can be used together with Linear Elastic Material, Nonlinear Elastic Material, and Hyperelastic Material. It is available with the Geomechanics Module. **Soil Plasticity** is available for 3D, 2D, and 2D axisymmetry.

When a **Soil Plasticity** node is present, an **Equivalent Plastic Strain** plot is available under **Result Templates**.

The failure criteria are described in the *Structural Mechanics Theory* chapter:

- Drucker–Prager Criterion
- Mohr–Coulomb Criterion
- Matsuoka–Nakai Criterion
- Lade–Duncan Criterion.

SOIL PLASTICITY

Select the **Failure criterion** — **Drucker–Prager**, **Mohr–Coulomb**, **Matsuoka–Nakai**, or **Lade–Duncan**.

Plastic Potential

Select the **Plastic potential** Q_p related to the flow rule — **Associated**, **Nonassociated**, or **Drucker–Prager**.

Equivalent Plastic Strain

Select how the **Equivalent plastic strain** ε_{pe} is computed — **Associated**, **von Mises**, or **User defined**. Enter a **User defined** value in the h_p field as needed. See Hardening Rule for details.

Drucker–Prager

In the standard Drucker–Prager formulation, the material parameters are given in terms of the coefficients α and k. Often, material data is expressed by the cohesion c and friction angle ϕ used in the Mohr–Coulomb criterion.

Select **Match to Mohr–Coulomb** — **Compressive meridian**, **Tensile meridian**, **Inscribe (plane strain)**, **Uniaxial (plane stress)**, **Biaxial (plane stress)**, or **User defined**. The default values

for the **Initial cohesion** c_0 and **Friction angle** ϕ are taken **From material**. For **User defined**, the values for the Drucker–Prager parameters k_0 and α and are taken **From material**.

When **Nonassociated** is selected in the **Plastic potential** list, the dilatation angle ψ replaces the friction angle ϕ in the plastic potential Q_p. The default value for the **Dilatation angle** ψ is taken **From material**. If not matched to the Mohr–Coulomb criterion, parameter α_Q defines the nonassociated plastic potential. It is by default taken **From material**.

If required, add a Cap and Cutoff subnode.

Mohr–Coulomb

When **Associated** is selected in the **Plastic potential** list, the default values for the **Initial cohesion** c_0 and **Friction angle** ϕ and are taken **From material**.

When **Nonassociated** is selected in the **Plastic potential** list, the dilatation angle ψ replaces the friction angle ϕ in the plastic potential Q_p. The default value for the **Dilatation angle** is taken **From material**.

When **Drucker–Prager** is selected in the **Plastic potential** list, a Drucker–Prager plastic potential with a dilatation angle ψ replaces the Mohr–Coulomb potential. Select the **Dilatation angle** — **From material**, **From friction angle**, or **User defined**.

If required, add a Cap and Cutoff subnode.

Matsuoka–Nakai

Select **Match to Mohr–Coulomb** — **Compressive meridian** or **User defined**. The default values for the **Initial cohesion** c_0 and **Friction angle** ϕ are taken **From material**. For **User defined**, the values for the Drucker–Prager parameters k_0 and α and are taken **From material**.

When **Nonassociated** is selected in the **Plastic potential** list, the dilatation angle ψ replaces the friction angle ϕ in the plastic potential Q_p. The default value for the **Dilatation angle** is taken **From material**.

When **Drucker–Prager** is selected in the **Plastic potential** list, a Drucker–Prager plastic potential with a dilatation angle ψ replaces the Matsuoka–Nakai potential. Select the **Dilatation angle** — **From material**, **From friction angle**, or **User defined**.

If not matched to the Mohr–Coulomb criterion, parameter α_Q defines the nonassociated plastic potential. It is by default taken **From material**.

If required, add a Cap and Cutoff subnode.

Lade–Duncan

Select **Match to Mohr–Coulomb** — **Compressive meridian** or **User defined**. The default values for the **Initial cohesion** c_0 and **Friction angle** ϕ are taken **From material**. For **User defined**, the values for the Drucker–Prager parameters k_0 and α and are taken **From material**.

When **Nonassociated** is selected in the **Plastic potential** list, the dilatation angle ψ replaces the friction angle ϕ in the plastic potential Q_p. The default value for the **Dilatation angle** is taken **From material**.

When **Drucker–Prager** is selected in the **Plastic potential** list, a Drucker–Prager plastic potential with a dilatation angle ψ replaces the Lade–Duncan potential. Select the **Dilatation angle** — **From material**, **From friction angle**, or **User defined**.

If not matched to the Mohr–Coulomb criterion, parameter α_Q defines the nonassociated plastic potential. It is by default taken **From material**.

If required, add a Cap and Cutoff subnode.

Isotropic Hardening

For all failure criteria, select the type of hardening model from the **Isotropic hardening model** list. See Isotropic Hardening for details.

- Select **Perfectly plastic** (ideal plasticity) if the material can undergo plastic deformation without any change in cohesion.
- For **Hardening function**, the isotropic **Hardening function** $c_h(\varepsilon_{pe})$ uses values **From material** or **User defined**. The cohesion is modified as

$$c = c_0 + c_h(\varepsilon_{pe})$$

  When **Match to Mohr–Coulomb** is set to **User defined**, the isotropic **Hardening Function** $k_h(\varepsilon_{pe})$ directly specifies the evolution of the strength parameter k as

$$k = k_0 + k_h(\varepsilon_{pe})$$

  - The above definitions imply that the hardening functions in the Material node must be zero at zero plastic strain. For example, $c = c_0$ when $\varepsilon_{pe} = 0$. With this option it is possible to enter any nonlinear curve. The hardening function can depend on more variables than the equivalent plastic strain, for example the temperature. Select **User defined** to enter any function of the equivalent plastic

strain ε_{pe}. The variable is named using the scheme `<physics>.epe`, for example, `solid.epe`.

See also Pressure-Dependent and Soil Plasticity in the *Structural Mechanics Theory* chapter.

- *Deep Excavation*: Application Library path **Geomechanics_Module/Soil/deep_excavation**
- *Flexible and Smooth Strip Footing on a Stratum of Clay*: Application Library path **Geomechanics_Module/Soil/flexible_footing**

NONLOCAL PLASTICITY MODEL

Nonlocal plasticity can be used to facilitate, for example, the modeling of material softening. Typical examples that involve material softening are finite-strain plasticity and soil plasticity. In these situations, standard (local) plasticity calculations reveal a mesh and topology dependence, where a mesh refinement fails to reproduce a physically sound solution. Nonlocal plasticity adds regularization to the equivalent plastic strain, thereby stabilizing the solution.

The default setting is **None**. Select **Implicit Gradient** to add nonlocal regularization to the equivalent plastic strain. Then, enter values or expressions for:

- **Length scale**, l_{int} — The regularization length scale for the equivalent plastic strain ε_{pe} should ideally be greater than the largest mesh element size.
- **Nonlocal coupling modulus**, H_{nl} — The value represents the penalization of the difference between the local and nonlocal variables. A larger value enforces the equivalent plastic strain ε_{pe} to be closer to the nonlocal equivalent plastic strain $\varepsilon_{pe,nl}$. A typical value would be around 10% the equivalent shear modulus of the material.

See also Nonlocal Plasticity in the *Structural Mechanics Theory* chapter.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog, and it is available with the **Implicit gradient** nonlocal plasticity model.

Select the shape function for the **Nonlocal equivalent plastic strain** $\varepsilon_{pe,nl}$ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The available options depend on the order of the displacement field.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The plastic potential includes a vertex where it intersects the hydrostatic axis. To improve robustness of the plasticity algorithm, smoothing is applied to remove these singularities.

Select the **Smoothing of plastic potential** — **Automatic**, **Manual tuning**, or **User defined** to control how much the smoothed plastic potential deviates from the original potential.

When **Manual tuning** or **User defined** is selected, enter the **Edge smoothing parameter** β when using Mohr–Coulomb plastic potential. The default value is $\beta = 0.99$.

When **Manual tuning** is selected, enter a value for the **Vertex smoothing multiplier** f_v. This multiplier scales the amount of smoothing at the apex given by the **Automatic** option.

The **User defined** option allows full control of the smoothing by entering a value for the **Vertex smoothing parameter** $\sigma_{v,off}$. A typical value would be around 10% of the initial cohesion for soil models.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** or **Backward Euler, damped** is selected, it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

When the **Backward Euler, damped** method is selected, the Newton's method is enhanced by line search iterations. Using this method can improve the robustness of the plasticity algorithm when the plastic potential or hardening model are highly nonlinear. When selected, it is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

See also Apex Smoothing, Vertex Smoothing, and Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Soil Plasticity**
Solid Mechanics > **Nonlinear Elastic Material** > **Soil Plasticity**
Solid Mechanics > **Hyperelastic Material** > **Soil Plasticity**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Soil Plasticity**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Soil Plasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Soil plasticity**

Cap and Cutoff

Use the **Cap and Cutoff** subnode to define additional contributions to the plasticity model defined in the parent Pressure-Dependent Plasticity, Soil Plasticity or Elastoplastic Soil Material node.

You can add a **Cap and Cutoff** node to the Elastoplastic Soil Material node when the Hardening Soil or the Hardening Soil Small Strain model is selected. In this case, the variable for the **Initial pressure limit** p_{c0} is called **Preconsolidation pressure**.

COMPRESSION CAP

Select the **Cap model** — **Elliptic**, **Planar**, **Cam-clay**, or **None**.

- For **Elliptic**, enter the **Initial pressure limit** p_{c0} and the **Initial ellipse centroid** p_{cc0}. The ellipse aspect ratio is then derived from the intersection of the ellipse with the criterion defined in the parent node.
- For **Planar**, enter the **Initial pressure limit** p_{c0}.
- For **Cam-clay**, enter the **Initial pressure limit** p_{c0}.

Select the **Hardening model** — **Perfectly plastic** (no hardening), **Linear**, **Exponential**, or **Hardening function**.

- When **Linear** is selected from the list, the default **Hardening modulus** K_c is taken **From material**.
- When **Exponential** is selected from the list, the default **Hardening modulus** K_c and the **Maximum volumetric plastic strain** $\varepsilon_{pvol,max}$ are taken **From material**.
- When **Hardening function** is selected from the list, the default **Hardening modulus** p_{ch} and the **Maximum volumetric plastic strain** $\varepsilon_{pvol,max}$ are taken **From material**.

Select the **Hardening variable** — **Volumetric plastic strain**, **Equivalent plastic strain**, or **User defined**.

See also Compression Cap in the *Structural Mechanics Theory* chapter.

TENSION CUTOFF

Select the **Cutoff model** — **Mean stress**, **Rankine**, or **None**.

When **Mean stress** is selected from the list, enter a value or an expression for the **Mean stress limit** $\sigma_{m,t}$. Use this value or expression to constrain the failure criterion model with a maximum allowed pressure in tension.

When **Rankine** is selected from the list, enter a value or an expression for the **Tensile strength** σ_{ts}. Use the tensile strength to constrain the failure criterion with a maximum principal tensile stress.

See also Tension Cutoff in the *Structural Mechanics Theory* chapter.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Smoothing** method to intersect the cap or cutoff criterion with the plasticity criterion — **Automatic**, **Manual tuning**, or **User defined**.

When **Manual tuning** is selected, it is possible to specify the relative transition zone where the cap or cutoff criterion intersects with the criterion specified the parent node. Enter the **Cap smoothing multiplier** f_{cap} and the **Cutoff smoothing multiplier** f_{cutoff}. Each multiplier scales the size of the transition zone for the **Automatic** option.

When **User defined** is selected, it is possible to specify the transition zone where the cap or cutoff criteria intersects with the failure criterion specified the parent node. Enter the **Cap smoothing parameter** $\sigma_{\text{c,off}}$ and the **Cutoff smoothing parameter** $\sigma_{\text{t,off}}$.

When the **Hardening model** for the cap is other than **Perfectly plastic**, select the **Allow cap softening** checkbox to soften the cap when loaded in tension.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Pressure-Dependent Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Nonlinear Elastic Material** > **Pressure-Dependent Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Hyperelastic Material** > **Pressure-Dependent Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Linear Elastic Material** > **Soil Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Nonlinear Elastic Material** > **Soil Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Hyperelastic Material** > **Soil Plasticity** > **Cap and Cutoff**
Solid Mechanics > **Elastoplastic Soil Material** > **Cap and Cutoff**

Ribbon

Physics tab with **Pressure-Dependent Plasticity**, **Soil Plasticity**, or **Elastoplastic Soil Material** node selected in the model tree:

Attributes > **Cap and Cutoff**

Concrete

The **Concrete** subnode includes three different material models for modeling concrete and other quasibrittle materials with similar properties:

- Coupled Damage–Plasticity

- Mazars Damage
- Perfect Plasticity

These concrete models can be used together with Linear Elastic Material and Nonlinear Elastic Material. It is available with the Geomechanics Module. **Concrete** is available for 3D, 2D, and 2D axisymmetry.

When a **Concrete** node is present, an **Equivalent Plastic Strain** plot is available under **Result Templates**.

CONCRETE MODEL

Use this section to define properties of the concrete material.

Select a **Material model** — **Coupled damage–plasticity**, **Mazars damage**, or **Perfect plasticity**, then follow the instructions below.

Coupled Damage–Plasticity

The Coupled damage–plasticity material model combines damage mechanics with a plasticity model to describe important characteristics of the mechanical response and failure of concrete subjected to multiaxial and cyclic loading conditions.

See also:

- Coupled Damage–Plasticity in the *Structural Mechanics Theory* chapter.
- Modeling Damage in the *Structural Mechanics Modeling* chapter.

Enter the uniaxial **Tensile strength** σ_{ts} and the **Compressive strength** σ_{cs}. These parameters are by default taken **From material**. When **User defined**, enter other values or expressions.

The **Biaxial compressive strength** σ_{bc} is defined as $\sigma_{bc} = 1.16\sigma_{cs}$ when set to **From default suggestion**. Other values can be used if set to **From Material** or **User defined**.

The **Compressive yield stress** σ_{cy} controls when plastic deformation starts. It is defined $\sigma_{cy} = 0.3\sigma_{cs}$ when set to **From default suggestion**. Other values can be used if set to **From Material** or **User defined**.

Tension softening

The settings in the Tension softening section controls the post-peak response of the model when loaded in tension. Select the type of softening function — **Linear**, **Bilinear**, **Cornelissen**, **Exponential**, **User–defined strain**, **User–defined displacement**, or **None**. Selecting **None** disables the tensile damage model. The **Bilinear** curve is used by default.

When **Linear**, **Bilinear**, **Cornelissen**, or **Exponential** is selected, specify how to set the **Tensile fracture energy** G_{ft}— **From material**, **From strain limit**, **From displacement limit**, or **User defined**. If necessary, also enter a value for the **Characteristic length** l_{ct}.

Using the **Cornelissen**, **Exponential**, or a **User defined** softening function requires solving an extra nonlinear equation to determine the tensile damage variable.

Compression Softening

The settings in the Compression softening section controls the post-peak response of the model when loaded in compression. Select the type of softening function — **Linear**, **Bilinear**, **Exponential**, **User–defined strain**, **User–defined displacement**, or **None**. Selecting **None** disables the tensile damage model. The **Linear** curve is used by default.

When **Linear**, **Bilinear**, or **Exponential** is selected, specify how to set the **Compressive fracture energy** G_{fc}— **From material**, **From strain limit**, **From displacement limit**, or **User defined**. By default **From strain limit** is used with a value equal to 1e-4. If necessary, also enter a value for the **Characteristic length** l_{cc}.

Enter a value for the **Ductility parameter** A_{s}. The parameter affects the softening response under multiaxial stress states. The default is 10.

Using the **Exponential** or a **User defined** softening function requires solving an extra nonlinear equation to determine the compressive damage variable.

Spatial Regularization Method

Select a spatial regularization method for the tensile damage model — **Crack band** or **None**. By default the **Crack band** method is used for the tensile response.

Select a spatial regularization method for the compressive damage model — **Crack band** or **None**. By default the no regularization is used in compression.

If the **Crack band** method is used in either tension or compression, specify the **Crack band calculation** — **Element volume/area** or **Element size**. The **Element volume/area** setting computes a representative crack band based on the mesh element type, whereas the **Element size** always picks the longest edge of the mesh element. See Spatial Regularization for details.

Mazars Damage

The Mazars damage model defines a strain–based scalar damage model that is a computationally lean alternative for modeling the failure of concrete during monotonic loading.

Select the type of **Equivalent strain** — **Mazars**, **Modified Mazars**, or **User defined**. See Mazars Damage for Concrete for details.

Enter the **Shear exponent** β, the default is set to 1.06.

Select the **Tensile damage evolution** — **Linear**, **Exponential**, **Polynomial**, **Multilinear**, **Mazars function**, or **User defined**.

- For **Linear**, **Exponential**, **Polynomial**, and **Multilinear**, enter the **Tensile strength** σ_{ts}, the default is to take the value **From material**. Also specify how to define the **Tensile fracture energy** G_{ft} — **From material**, **From strain limit**, **From displacement limit**, or **User defined**. The default is to take the value **From material**.If necessary, also enter a value for the **Characteristic length** l_{ct}. For the Multilinear option, also enter a value for the **Shape factor** λ, the default is set to 0.5.
- For **Mazars function**, enter the **Tensile strain threshold** ε_{0t}, and the **Tensile damage parameters** A_t and B_t.
- For **User defined**, enter an expression for the **Tensile damage evolution** function $d_t(\kappa)$.

Select the **Compressive damage evolution** — **Mazars function** or **User defined**.

- For **Mazars function**, enter the **Compressive strain threshold** ε_{0c}, and the **Compressive damage evolution parameters** A_c and B_c.
- For **User defined**, enter an expression for the **Compressive damage evolution** function $d_c(\kappa)$.

Select the type of **Spatial regularization method** — **None**, **Crack band**, or **Implicit gradient**. See Spatial Regularization for details.

- For the **Crack band** method select the type of **Crack band calculation** — **Element volume/area** or **Element size**.
- For the **Implicit gradient** method enter the **Length scale** l_{int}. If the **Fracture energy per area** was selected as **Strain softening input**, enter also the **Characteristic size of the damage dissipation zone** h_{dmg}. The **Implicit gradient** method is available in the Solid Mechanics and Layered Shell interfaces.

Select the type of **Viscous regularization method** — **None** or **Delayed Damage**.

For the **Delayed damage** method, enter the **Characteristic time** τ. The Delayed damage method is intended for time-dependent studies, and adds no contributions for other study types. See Viscous Regularization for details.

Perfect Plasticity

The Perfect plasticity material model defines a yield surface suitable to capture the failure of concrete under multiaxial loading conditions. As the name implies, no hardening is included and the purpose of the model is thus mainly to identify if the strength of the material is reached.

Select a **Yield function** — **Bresler–Pister**, **Willam–Warnke**, or **Ottosen**. The default values for the material parameters are taken **From material**. For **User defined** choices, enter other values or expressions.

When **Bresler–Pister** or **Willam–Warnke** is selected, enter the **Tensile strength** σ_{ts}, **Compressive strength** σ_{cs}, and **Biaxial compressive strength** σ_{bc}.

When **Ottosen** is selected enter the **Compressive strength** σ_{cs}, Ottosen parameters a and b, **Size factor** k_1, and **Shape factor** k_2.

The failure criteria are described in the theory section:

- Bresler–Pister Criterion
- Willam–Warnke Criterion
- Ottosen Criterion.

TENSION CUTOFF

This section is available for the **Perfect plasticity** material model. If required, select either **None** or **Principal stress cutoff**.

When **Principal stress cutoff** is selected from the list, enter a value or an expression for the **Maximum tensile stress** σ_t. Use this to constrain the concrete model with an extra yield surface, which limits the maximum principal tensile stress.

See also Tension Cutoff in the *Structural Mechanics Theory* chapter.

ADVANCED PARAMETERS

This section is available for the **Coupled damage–plasticity** and **Mazars damage** material models.

For both material models, enter the **Maximum damage**. This settings controls the residual stiffness of the material when fully damaged kept for numerical stability. The default is 1-1e-5.

When the **Coupled damage–plasticity** model is selected, this section allows you to enter values for several advanced model parameters modify the behavior of the model. In the majority of cases the default values results in a good prediction.

- The **Dilatancy parameter** D_f controls the amount of volumetric plastic deformation during compressive loading. It can be considered as the ratio of the absolute values of lateral and axial plastic strains during uniaxial compression. The default value is 0.85.
- The **Hardening modulus** H_p controls the evolution of the yield surface during plastic flow. Its value must be set between 0 and σ_{cy}/σ_{cs}. The default value is 0.01.
- The **Hardening ductility parameter** A_h, B_h, C_h, and D_h controls the evolution of the scalar hardening variable and can be used to adjust the ductility of the yield surface for different loadings. The default values are 0.08, 0.003, 2, and 1e-6.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The plastic potential of the **Coupled damage–plasticity** model includes an apex where it intersects the hydrostatic axis. To improve robustness of the plasticity method, smoothing is applied to remove these singularities. Select the **Smoothing of the plastic potential** — **Automatic**, **Manual tuning**, or **User defined**, to control how much the smooth

plastic potential deviates from the original plastic potential. When **Manual tuning** is selected, enter a value for the **Vertex smoothing multiplier** f_v. This multiplier scales the amount of smoothing of the **Automatic** option. The **User defined** option allows full control of the smoothing by entering a value for the **Vertex smoothing parameter** $\sigma_{v,off}$.

Select the **Local method** to solve the plasticity problem — **Automatic**, **Backward Euler**, or **Backward Euler, damped**. When **Backward Euler** is selected, it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

The **Backward Euler, damped** option is only available when the **Coupled damage–plasticity** model is selected. When selected, it enhances the Newton's method by line search iterations. It is possible to specify the **Maximum number of line search iterations**. The default value is 4 iterations.

See also the Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Concrete**
Solid Mechanics > **Nonlinear Elastic Material** > **Concrete**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Concrete**

Ribbon

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic Material** node selected in the model tree:

Attributes > **Concrete**

Rocks

In the **Rocks** subnode you define the properties for modeling materials with failure criteria representative of rocks. This material model can be used together with Linear Elastic Material and Nonlinear Elastic Material. It is available with the Geomechanics Module. The **Rocks** subnode is available for 3D, 2D, and 2D axisymmetry.

When a **Rocks** node is present, an **Equivalent Plastic Strain** plot is available under **Result Templates**.

The failure criteria are described in the *Structural Mechanics Theory* chapter:

- Original Hoek–Brown Criterion
- Generalized Hoek–Brown Criterion

ROCK MODEL

Select the **Material model** — **Original Hoek–Brown** or **Generalized Hoek–Brown**. The default values for the material parameters are taken **From material**. For **User defined** choices, enter other values or expressions.

Original Hoek–Brown

The defaults for the **Compressive strength** σ_{cs}, **Hoek–Brown parameter** m, and **Hoek–Brown parameter** s are taken **From material**.

Generalized Hoek–Brown

The defaults for the **Compressive strength** σ_{cs}, **Geological strength index** GSI, **Disturbance factor** D, and **Intact rock parameter** m_i are taken **From material**.

TENSION CUTOFF

If required, select either **None** or **Principal stress cutoff**.

When **Principal stress cutoff** is selected from the list, enter a value or an expression for the **Maximum tensile stress** σ_t. Use this to constrain the rock model with an extra yield surface, which limits the maximum principal tensile stress.

See also Tension Cutoff in the *Structural Mechanics Theory* chapter.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Local method** to solve the plasticity problem — **Automatic** or **Backward Euler**. When **Backward Euler** is selected, it is possible to specify the maximum number of iterations and the relative tolerance used to solve the local plasticity equations. Enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local plasticity equations. The default value is 25 iterations.
- **Relative tolerance**. To check the convergence of the local plasticity equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value. The default value is 1e-6.

See also the Numerical Integration Algorithm section in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Rocks**

Solid Mechanics > **Nonlinear Elastic Material** > **Rocks**

Ribbon

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic Material** node selected in the model tree:

Attributes > **Rocks**

Fiber

Add one or more **Fiber** nodes to add stiffness in specified directions to a solid material. The **Fiber** feature can be used together with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material**. The assumption is that the volume of the fibers is small when compared to the base material.

The **Fiber** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). The **Fiber** subnode is available for 3D, 2D, and 2D axisymmetry.

FIBER MODEL

Linear Elastic Material or Nonlinear Elastic Material

Select a **Material model** — **Linear elastic** or **Uniaxial data**.

The option **Uniaxial data** is not available in the Solid Mechanics, Explicit Dynamics interface.

From the **Material** list, select the **Domain material** (the default) or any other material to define the fiber's properties. In most cases, you would use the domain material for the base material, and additional **Material** nodes without domain selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- For **Linear elastic** enter the **Young's modulus** E_{fiber} and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions.
- For **Uniaxial data** enter the **Uniaxial stress function** $\sigma_{\text{ax,a}}$ and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions. The default user defined expression for $\sigma_{\text{ax,a}}$ is the linear function `210[GPa]*<item>.eea`, where the variable `<item>.eea` is the elastic strain in the fiber direction, and `<item>` corresponds to the tag of the **Fiber** node, for example `solid.lemm1.fib1`. See Fibers for Linear and Nonlinear Elastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

Hyperelastic Material

Select a **Material model** — **Holzapfel–Gasser–Ogden**, **Linear elastic**, **Uniaxial data**, or **User defined**. The default values of the required model parameters are taken **From material**.

From the **Material** list, select the **Domain material** (the default) or any other material to define the fiber properties. In most cases, you would use the domain material for the base material, and additional **Material** nodes without domain selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- The **Holzapfel–Gasser–Ogden** model defines an exponential strain energy density W_{fiber}. The values for the **Fiber stiffness** k_1, **Model parameter** k_2, and **Fiber dispersion** k_3 are as a default taken **From material**. Select **User defined** to enter other values or expressions.
- Select **Linear elastic** or **Uniaxial data** to describe the fiber stiffness. See Linear Elastic Material or Nonlinear Elastic Material for details.
- If **User defined** is selected from the list, select a **Material symmetry** — **Isotropic**, **Transversely isotropic**, or **Anisotropic**.

  Depending on the choice, different stretch invariants are available to define the **Fiber strain energy density** W_{fiber}. The default expression for W_{fiber} is the function `0[Pa]*(<item>.IaCIe-1)`, where the variable `<item>.IaCIe` is the isochoric stretch invariant in the fiber direction, and `<item>` corresponds to the tags of the **Fiber** node, for example `solid.hmm1.fib1`. See Fibers for Hyperelastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

For the **Holzapfel–Gasser–Ogden** model, select the **Use isochoric deformation** checkbox to account for the fiber incompressibility. See Compressible Fibers in the *Structural Mechanics Theory* chapter for details.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

DISTRIBUTION AND ORIENTATION

Enter the fiber **Volume fraction** v_{fiber}. To be consistent with the underlying assumptions, it should not exceed a few percent.

Select a **Fiber orientation** from the list. The available choices are the axis directions of the coordinate system selected in the **Coordinate System Selection** section, or **User**

defined. For **User defined**, enter a **Direction**. The direction vector **a** is interpreted in the selected local coordinate system.

See also Distributed Fiber Models in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Fiber**
Solid Mechanics > **Nonlinear Elastic Material** > **Fiber**
Solid Mechanics > **Hyperelastic Material** > **Fiber**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Fiber**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Fiber**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Fiber**

Thermal Expansion (Fiber)

When used together with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** it is possible to model thermal expansion in the fibers. The assumption is that the volume of fibers is small when compared to the base material, and that thermal expansion (or contraction) occurs in the fiber direction only.

For **Holzapfel–Gasser–Ogden** and **User defined** fibers under **Hyperelastic Material**, the thermal expansion subnode is disabled when the checkbox **Contribute to total stress** is selected.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains in the fibers. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the temperature locally.

The fiber **Temperature** T is by default obtained from a **Common model input**. You can also select an existing temperature variable from a heat transfer interface, if any

temperature variables exist, or manually enter a value or an expression by selecting **User defined**.

THERMAL EXPANSION PROPERTIES

The settings in this section depend slightly on whether the parent **Fiber** node is a domain feature or a boundary feature.

Domain Feature

From the **Material** list, select the **Domain material** or any other material to define the fiber's thermal expansion properties. In most cases, you would use the domain material for the base material, and additional **Material** nodes without domain selection as the fiber material.

Boundary Feature (Membrane or Thin Layer)

From the **Material** list, select the **Boundary material** or any other material to define the fiber's thermal expansion properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without domain selection as the fiber material.

Select an **Input type** to specify how the thermal strains in the fibers should be modeled. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain in the fibers is given by

$$\varepsilon_{\text{th,fib}} = \alpha(T - T_{\text{ref}})$$

Here, the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain in the fibers is given by

$$\varepsilon_{\text{th,fib}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\text{t}}(\tau)d\tau\right) - 1$$

where α_{t} is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter explicitly the thermal strain in the fibers dL as function of temperature.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, enter the coefficient of the thermal expansion or the thermal strain in the fiber direction **a**, since thermal expansion is considered in the fiber direction only.

In all cases, the contribution to the thermal strain tensor is defined from the thermal strain in the fibers $\varepsilon_{th,fib}$, and the fiber direction $\mathbf{a}$ as

$$\varepsilon_{th} = \varepsilon_{th,fib}\mathbf{a} \otimes \mathbf{a}$$

See also Thermal Expansion in Fibers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics > **Nonlinear Elastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics > **Hyperelastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics > **Thin Layer** > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics > **Thin Layer** > **Nonlinear Elastic Material Fiber** > **Thermal Expansion**
Solid Mechanics > **Thin Layer** > **Hyperelastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Fiber** > **Thermal Expansion**
Membrane > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**
Membrane > **Nonlinear Elastic Material** > **Fiber** > **Thermal Expansion**
Membrane > **Hyperelastic Material** > **Fiber** > **Thermal Expansion**

Ribbon

Physics tab with **Fiber** node selected in the model tree:

Attributes > **Thermal Expansion**

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

The **Thermal Expansion** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is present when **Thermal Expansion** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Thermal Expansion (for Materials) node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Thermal Expansion (for Materials) node in the Membrane chapter.

This section is only present in the Layered Shell interface, where it is described in the documentation for the Thermal Expansion (for Materials) node.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the temperature locally.

The **Temperature** T is by default obtained from a **Common model input**. You can also select an existing temperature variable from a heat transfer interface (for example, **Temperature (ht/sol1)**), if any temperature variables exist, or manually enter a value or an expression by selecting **User defined**.

Thermal strains are proportional to the temperature, while structural strains are computed from the gradient of the displacement field. It is a good practice to match the discretization order of thermal and structural strains.

When adding a **Thermal Expansion** subnode in a Layered Shell or Membrane interface, and the temperature field is computed by another physics interface (often the Heat Transfer in Shells interface); use a discretization one order lower for the temperature field than what is used for the displacement field.

A consistent strain discretization is automatically set up for the Solid Mechanics interface.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

THERMAL EXPANSION PROPERTIES

Select an **Input type** to specify how the thermal strain is entered. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

here, the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

A heat source term will be created by this node. It can be accessed from a **Thermoelastic Damping** node in a heat transfer interface in order to incorporate the reversed effect that heat is produced by changes in stress. The heat source term is only present when Structural Transient Behavior is set to **Include inertial terms**.

- Temperature-Dependent Material Data
- Thermoelastic Damping
- Entropy and Thermoelasticity

Thermal Stresses in a Layered Plate: Application Library path
Structural_Mechanics_Module/Thermal-Structure_Interaction/layered_plate
MEMS_Module/Actuators/layered_plate

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Thermal Expansion**
Solid Mechanics > **Nonlinear Elastic Material** > **Thermal Expansion**
Solid Mechanics > **Elastoplastic Soil Material** > **Thermal Expansion**
Solid Mechanics > **Hyperelastic Material** > **Thermal Expansion**
Solid Mechanics > **Piezoelectric Material** > **Thermal Expansion**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Thermal Expansion**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Thermal Expansion**
Layered Shell > **Linear Elastic Material** > **Thermal Expansion**
Layered Shell > **Hyperelastic Material** > **Thermal Expansion**
Layered Shell > **Piezoelectric Material** > **Thermal Expansion**
Membrane > **Linear Elastic Material** > **Thermal Expansion**
Membrane > **Linear Elastic Material, Layered** > **Thermal Expansion**
Membrane > **Nonlinear Elastic Material** > **Thermal Expansion**
Membrane > **Hyperelastic Material** > **Thermal Expansion**
Multibody Dynamics > **Linear Elastic Material** > **Thermal Expansion**
Solid Rotor > **Linear Elastic Material** > **Thermal Expansion**
Solid Rotor, Fixed Frame > **Linear Elastic Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material**, **Hyperelastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Piezoelectric Material**, or **Piezomagnetic Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_{\mathrm{h}} M_m (c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, M_m is the molar mass, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration.

The **Hygroscopic Swelling** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is present when **Hygroscopic Swelling** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Hygroscopic Swelling node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Hygroscopic Swelling node in the Membrane chapter.

MODEL INPUTS

From the **Concentration** *c* list, select an existing concentration variable from another physics interface, if any concentration variables exist. For **User defined** enter a value or an expression for the concentration. The unit for the input depends on the setting of

Concentration type in the Hygroscopic Swelling Properties section. Only concentration variables having the chosen physical dimension are available in the **Concentration** list.

Hygroscopic strains are proportional to the concentration, while structural strains are computed from the gradient of the displacement field. It is a good practice to match the discretization order of hygroscopic and structural strains.

When adding a **Hygroscopic swelling** subnode in a Layered Shell or Membrane interface, and the concentration field is computed by another physics interface; use a discretization one order lower for the concentration field than what is used for the displacement field.

A consistent strain discretization is automatically set up for the Solid Mechanics interface.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref}. This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as the **Concentration type**, enter also the **Molar mass** of the fluid, M_{m}. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** from the list to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h}. The default value for the **User defined** case is `1.5e-4` m^3/kg. When a nonisotropic coefficient of hygroscopic swelling is used, the axis orientations are given by the coordinate system selection in the parent node.

The **Include moisture as added mass** checkbox is selected by default. When selected, the mass of the fluid is included in a dynamic analysis, and when using mass proportional loads. It will also contribute when computing mass properties.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Hygroscopic Swelling**

Solid Mechanics > **Nonlinear Elastic Material** > **Hygroscopic Swelling**
Solid Mechanics > **Hyperelastic Material** > **Hygroscopic Swelling**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Hygroscopic Swelling**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Hygroscopic Swelling**
Layered Shell > **Linear Elastic Material** > **Hygroscopic Swelling**
Layered Shell > **Hyperelastic Material** > **Hygroscopic Swelling**
Membrane > **Linear Elastic Material** > **Hygroscopic Swelling**
Membrane > **Linear Elastic Material, Layered** > **Hygroscopic Swelling**
Membrane > **Nonlinear Elastic Material** > **Hygroscopic Swelling**
Membrane > **Hyperelastic Material** > **Hygroscopic Swelling**

Ribbon
Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Intercalation Strain

Intercalation strains are internal volumetric strains that occur as a result of the insertion of species (atoms) into interstitial sites in a host electrode material, for instance, intercalation of lithium atoms in graphite electrodes in lithium-ion batteries.

The volumetric strain due to intercalation is given by a function $dV(c) = \Delta V/V_0$, where ΔV is the change in volume due to intercalation with respect to a reference volume V_0. In case of small strains, the intercalation strain tensor reads

$$\varepsilon_{\mathrm{ic}} = \frac{1}{3}(dV(c) - dV(c_{\mathrm{ref}}))I$$

where I is the identity tensor and c_{ref} is the value of the concentration for which no volumetric strain due to intercalation occurs. For geometrically nonlinear studies, the intercalation strain is represented by the deformation gradient

$$F_{\mathrm{ic}} = \left(\frac{1 + dV(c)}{1 + dV(c_{\mathrm{ref}})}\right)^{\frac{1}{3}} I$$

The **Intercalation Strain** subnode is only available with the Battery Design Module, see www.comsol.com/products/specifications/).

INTERCALATION STRAIN

Select how to define the **Volumetric strain** dV — **From material** or **User defined**.

When **From material** is selected, the material property **Volumetric strain** is used. The value of the selected **Concentration** is then passed to the material using the input in the **Model Input** section. Add the **Reference volumetric strain** dV_{ref} for which the strain tensor ε_{ic} should evaluate to zero or the deformation gradient $\boldsymbol{F}_{\text{ic}}$ should evaluate to one, respectively.

For **User defined**, enter an expression for the volumetric strain, $dV(c)$, as a function of a concentration variable to satisfy

$$\varepsilon_{\text{ic}} = \frac{1}{3} dV(c) \boldsymbol{I}$$

or

$$\boldsymbol{F}_{\text{ic}} = (1 + dV(c))^{\frac{1}{3}} \boldsymbol{I}$$

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Intercalation Strain**
Solid Mechanics > **Nonlinear Elastic Material** > **Intercalation Strain**
Solid Mechanics > **Hyperelastic Material** > **Intercalation Strain**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Intercalation Strain**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to several material models, in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense, but rather a contribution to the constitutive relation.

The **Initial Stress and Strain** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

SHELL PROPERTIES

This section is only present when **Initial Stress and Strain** is used in the Layered Shell interface. See the documentation for the Initial Stress and Strain node in the Layered Shell chapter.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. The given initial stresses and strains are interpreted in this system.

INITIAL STRESS AND STRAIN

Enter values or expressions for the **Initial stress** $\mathbf{S}_0$ and **Initial strain** ε_0. For both, enter the diagonal and off-diagonal components (based on space dimension):

- For a 3D **Initial stress** model, diagonal components $\mathbf{S}_{0x}$, $\mathbf{S}_{0y}$, and $\mathbf{S}_{0z}$ and off-diagonal components $\mathbf{S}_{0xy}$, $\mathbf{S}_{0yz}$, and $\mathbf{S}_{0xz}$, for example.
- For a 3D **Initial strain** model, diagonal components ε_{0x}, ε_{0y}, and ε_{0z} and off-diagonal components ε_{0xy}, ε_{0yz}, and ε_{0xz}, for example.

In a geometrically nonlinear analysis, the stresses should be interpreted as second Piola–Kirchhoff stresses, and the strains should be interpreted as Green–Lagrange strains.

- For details about initial stresses and strains, see Inelastic Strain Contributions and Initial Stress and Strain.
- For details about the different strain measures, see Deformation Measures.
- For details about the different stress measures, see Defining Stress.

For an example of adding initial stresses, see *Prestressed Micromirror*: Application Library path **MEMS_Module/Actuators/micromirror**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Initial Stress and Strain**
Solid Mechanics > **Nonlinear Elastic Material** > **Initial Stress and Strain**
Solid Mechanics > **Elastoplastic Soil Material** > **Initial Stress and Strain**
Solid Mechanics > **Piezoelectric Material** > **Initial Stress and Strain**
Solid Mechanics > **Piezomagnetic Material** > **Initial Stress and Strain**
Solid Mechanics > **Shape Memory Alloy** > **Initial Stress and Strain**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Initial Stress and Strain**
Layered Shell > **Linear Elastic Material** > **Initial Stress and Strain**
Multibody Dynamics > **Linear Elastic Material** > **Initial Stress and Strain**
Solid Rotor > **Linear Elastic Material** > **Initial Stress and Strain**
Solid Rotor, Fixed Frame > **Linear Elastic Material** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Piezoelectric Material**, **Piezomagnetic Material**, or **Shape Memory Alloy** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode to several material models, in order to specify an additional stress contribution which is not part of the constitutive relation. The external stress can be added to the total stress tensor, or act only as an extra load contribution.

The **External Stress** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor.

This section is only present when **External Stress** is used in the Layered Shell interface. See the documentation for the External Stress node in the Layered Shell chapter.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor (Material)**, **Stress tensor (Spatial)**, **Stress tensor (Nominal)**, **Pore pressure**, or **In situ stress**.

- When **Stress tensor (Material)** is selected, you enter the external stress in the form of a second Piola–Kirchhoff stress tensor. The **External stress tensor** list will contain all stress tensors announced by any physics interface, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** $\mathrm{S}_{\mathrm{ext}}$ as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** — to determine the effect of the contribution.
- When **Stress tensor (Spatial)** is selected, you enter the external stress in the form of a Cauchy stress tensor. The components are interpreted in the selected coordinate system. Depending on the properties of the tensor, you can enter the data for the **External stress tensor** σ_{ext} as **Isotropic**, **Diagonal**, or **Symmetric**. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** — to determine the effect of the contribution.
- When **Stress tensor (Nominal)** is selected, you enter the external stress in the form of a first Piola–Kirchhoff stress tensor. The components are interpreted in the selected coordinate system. You can enter the data for the **External stress tensor** $\mathrm{P}_{\mathrm{ext}}$ as a full tensor that represents current force per undeformed area. The contribution acts as external load.
- When **Pore pressure** is selected, the **Absolute pressure** list will always contain the entry **User defined**, in which case you manually enter a value or an expression for the absolute pressure p_{A}. If there are other physics interfaces (like **Darcy's Law**) that compute and announce a pressure variable, such variables are also present in the list. You can also enter a **Reference pressure level** p_{ref}, which is the pressure level at which the pore pressure does not give any stress contribution. Enter the **Biot–Willis coefficient** α_{B} to specify the fraction of the pore pressure to use. As a default, its value

is taken **From material**. Choose **User defined** to enter another value or expression. When using **Pore pressure**, there is no contribution to the stress tensor, the only effect of the pressure is as a load.

- When **In situ stress** is selected, you enter the external stress in the form of Cauchy stress tensor. The tensor components are interpreted in the selected coordinate system. You can enter the data for the **External stress tensor** σ_{ext} as **Isotropic**, **Diagonal**, or **Symmetric**. When using **In situ stress**, there are both contributions to the stress tensor, as well as a body load. The **In situ stress** option requires the Geomechanics Module.

- Selecting a stress tensor announced by the same physics interface as where the **External Stress** node is added, will result in an error (‘`Circular variable dependency detected`’). This operation would imply that the computed stress depends on itself.
- You can use a stress tensor from the same physics interface, but a previous solution step. Select the **User defined** input type and enter expressions where the withsol operator is used to point to the intended solution.
- Since all stress tensor representations coincide in a geometrically linear analysis, **Stress tensor (Spatial)** is needed only in the case of a geometrically nonlinear analysis. The stress tensor is entered using a Cauchy stress tensor representation, and is internally transformed to a second Piola–Kirchhoff stress tensor.

The **External stress** subnode can be used to account for the effect of pore pressure in a porous material in a Layered Shell interface.

When the pore pressure distribution is computed by another physics interface (often the Layered Darcy’s Law interface), use a discretization for the fluid pressure that is one order lower than what is used for the displacement field. The discretization order of stresses in the structural problem will then be consistent with that of the added pressure.

A consistent strain discretization is automatically set up for the Solid Mechanics interface.

- For theory, see External Stress.
- For details about the different stress measures, see Defining Stress.

For an example of how the **External Stress** node is used to describe in situ stresses, see *Deep Excavation*: Application Library path **Geomechanics_Module/Soil/deep_excavation**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **External Stress**
Solid Mechanics > **Nonlinear Elastic Material** > **External Stress**
Solid Mechanics > **Elastoplastic Soil Material** > **External Stress**
Solid Mechanics > **Hyperelastic Material** > **External Stress**
Solid Mechanics > **Shape Memory Alloy** > **External Stress**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **External Stress**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **External Stress**
Layered Shell > **Linear Elastic Material** > **External Stress**
Layered Shell > **Hyperelastic Material** > **External Stress**
Membrane > **Linear Elastic Material** > **External Stress**
Membrane > **Nonlinear Elastic Material** > **External Stress**
Membrane > **Layered Elastic Material** > **External Stress**
Membrane > **Hyperelastic Material** > **External Stress**
Solid Rotor > **Linear Elastic Material** > **External Stress**
Solid Rotor, Fixed Frame > **Linear Elastic Material** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Hyperelastic Material**, or **Shape Memory Alloy** node selected in the model tree:

Attributes > **External Stress**

External Strain

The **External Strain** subnode allows you to provide inelastic strain contributions to the material models Linear Elastic Material, Nonlinear Elastic Material, Hyperelastic Material, and Shape Memory Alloy on a variety of formats, including using external coded functions.

The **External Strain** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present when **External Strain** is used in the Layered Shell interface. See the documentation for the External Strain node in the Layered Shell chapter.

COORDINATE SYSTEM SELECTION

Select a **Coordinate system**. All inputs that you give below are interpreted in the given coordinate system. This also implies that external code should assume that strains are passed with its local orientations. Deformation gradients are rotated by the local system in both indices.

EXTERNAL STRAIN

Select the type of **Strain input** — **External material**, **Strain tensor**, **Deformation gradient**, **Deformation gradient, inverse**, **Stretches**, or **Volumetric strain**.

- The option **Strain tensor** is not available for a **Hyperelastic Material**.
- The option **External material** is not available in the Layered Shell interface.

External Material

For **External material**, the computation of an additional inelastic strain contribution is delegated to external code that has been compiled into a shared library. External libraries must first be imported into an **External Material** node under **Global Definitions** > **Materials**.

Select an **External material** from the list of compatible external materials added under **Global Definitions** > **Materials**.

For a material to be compatible with this **External Strain** subnode, its **Interface type** must be set to a type whose required input quantities are all defined in this node. Allowed required inputs include Green–Lagrange strains, the deformation gradient, second Piola–Kirchhoff stress as well as all standard model inputs. Select **Inelastic**

residual quantity — **Strain** or **Deformation gradient** to define the type of quantity by which you want communicate with the external code.

- If the **Interface type** in the **External material** node is **Inelastic residual strain**, then select **Strain**.
- If the **Interface type** in the **External material** node is **Inelastic residual deformation**, then select **Deformation gradient**. This is the only option for a hyperelastic material.

Strain Tensor

For **Strain tensor**, enter an inelastic strain contribution ε_{ext}. From the **Strain tensor** list, you can choose **User defined** or any strain tensor that is announced by another physics interface. If you select **User defined**, enter values or expressions for the upper diagonal part of the symmetric strain tensor.

Deformation Gradient

For **Deformation gradient**, enter an inelastic deformation gradient contribution $\boldsymbol{F}_{ext}$. From the **External deformation gradient** list, you can choose **User defined** or any deformation gradient tensor that is announced by another physics interface. If you select **User defined**, enter values or expressions for the components of the deformation gradient tensor.

Deformation Gradient, Inverse

For **Deformation gradient, inverse**, enter an inelastic inverse deformation gradient contribution $\boldsymbol{F}_{ext}^{-1}$. From the **External deformation gradient inverse** list, you can choose **User defined** or any inverse deformation gradient tensor that is announced by another physics interface. If you select **User defined**, enter values or expressions for the components of the inverse deformation gradient tensor.

Stretches

For **Stretches**, enter values or expressions for the stretches in the coordinate system selected in the Coordinate System Selection section. Entering data on this form is convenient for some simple geometries and strain states.

Volumetric strain

For **Volumetric strain**, enter a value or expression for the external volumetric strain contribution dV_{ext}.

- For theory, see External Strain.
- For details about the different strain measures, see Deformation Measures.

The **External Strain** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

- Selecting a tensor announced by the same physics interface as where the **External Strain** node is added, may result in an error ("`Circular variable dependency detected`"). This operation is usually meaningless.
- You can use a tensor from the same physics interface, but from a previous solution step. Select the **User defined** input type and enter expressions where the withsol operator is used to point to the intended solution.
- If you select **Deformation gradient** form the **Strain input** list, and the **Additive strain decomposition** checkbox is selected in the parent material node during a geometrically nonlinear study step, the Green–Lagrange strain tensor is computed from

$$\varepsilon_{\text{ext}} = \frac{1}{2}(F_{\text{ext}}^T F_{\text{ext}} - I)$$

- If you select **Deformation gradient** form the **Strain input** list when engineering strains are expected, the engineering strain tensor is computed from

$$\varepsilon_{\text{ext}} = \frac{1}{2}(F_{\text{ext}} + F_{\text{ext}}^T) - I$$

  This is the case if the study step is geometrically linear, or when the **Geometrically linear formulation** checkbox is selected in the parent material node.
- If you select **Strain tensor** from the **Strain input** list, and the parent material node operates with multiplicative strain decomposition, the external strain is converted into a deformation gradient using the infinitesimal strain assumption

$$F_{\text{ext}} = I + \varepsilon_{\text{ext}}$$

See also

- Using External Materials
- External Material and Working with External Materials in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **External Strain**
Solid Mechanics > **Nonlinear Elastic Material** > **External Strain**
Solid Mechanics > **Hyperelastic Material** > **External Strain**
Solid Mechanics > **Shape Memory Alloy** > **External Strain**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **External Strain**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **External Strain**
Shell > **Hyperelastic Material, Layered** > **External Strain**
Layered Shell > **Linear Elastic Material** > **External Strain**
Layered Shell > **Hyperelastic Material** > **External Strain**
Solid Rotor > **Linear Elastic Material** > **External Strain**
Solid Rotor, Fixed Frame > **Linear Elastic Material** > **External Strain**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Hyperelastic Material**, or **Shape Memory Alloy** node selected in the model builder tree:

Attributes > **External Strain**

Inelastic Strain Rate

The **Inelastic Strain Rate** node allows to enter inelastic strain contributions to the **Linear Elastic Material**, **Nonlinear Elastic Material**, and **Hyperelastic Material** models on a variety of formats. As opposed to the External Strain feature, the inelastic contribution is given as a strain rate, and the inelastic contribution is added after integrating this rate in time.

The **Inelastic Strain Rate** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). The node is available for 3D, 2D, and 2D axisymmetry.

SHELL PROPERTIES

This section is present when **Inelastic Strain Rate** is used as a subnode to:

- **Linear Elastic Material** or **Hyperelastic Material** in the Layered Shell interface. See the documentation for the **Inelastic Strain Rate** node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** in the Shell interface. See the documentation for the Inelastic Strain Rate node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Inelastic Strain Rate node in the Membrane chapter.

COORDINATE SYSTEM SELECTION

Select a **Coordinate system**. All inputs are interpreted in the given coordinate system. This also implies that external code should assume that strains are passed with its local orientations. Deformation gradients are rotated by the local system in both indices.

INELASTIC STRAIN

Select the type of **Strain input** — **Strain tensor**, **Deformation gradient**, **Deformation gradient, inverse**, or **Stretches**.

- The option **Strain tensor** is not available for a **Hyperelastic Material** or **Layered Hyperelastic**.
- The options **Deformation gradient**, **Deformation gradient, inverse**, and **Stretches** are not available with **Linear Elastic Material, Layered** in the Shell interface.

INITIAL VALUE

Enter an initial value for the inelastic quantity as specified by the **Strain input** setting.

This section is not present when **Inelastic Strain Rate** is used with:

- The Layered Shell interface.
- **Linear Elastic Material, Layered** and **Hyperelastic Material, Layered** in the Shell interface.
- **Linear Elastic Material, Layered** in the Membrane interface.

In the above cases, the domain ODEs method is always used.

Select a **Method** — **Automatic**, **Backward Euler**, or **Domain ODEs**.

Automatic

The **Automatic** option corresponds to the backward Euler method except with predefined settings for the Newton loop used to solve the local equations.

Backward Euler

For the **Backward Euler** method, enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iterations in the Newton loop when solving the local time-dependent equations.
- **Absolute tolerance**. To check the convergence of the local equations based on the step size in the Newton loop.
- **Relative tolerance**. To check the convergence of the local equations based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value.
- **Residual tolerance**. To check the convergence of the local equations based on the residual of each equation.

If both a step size and residual convergence check is requested, it is sufficient that either of the conditions are fulfilled. Setting either the **Absolute tolerance** and **Relative tolerance** or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all are set to zero.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of these fields can affect the convergence of the overall solution

See also Inelastic Strain Rate in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Inelastic Strain Rate**
Solid Mechanics > **Nonlinear Elastic Material** > **Inelastic Strain Rate**
Solid Mechanics > **Hyperelastic Material** > **Inelastic Strain Rate**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Inelastic Strain Rate**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Inelastic Strain Rate**
Shell > **Linear Elastic Material, Layered** > **Inelastic Strain Rate**
Shell > **Hyperelastic Material, Layered** > **Inelastic Strain Rate**
Layered Shell > **Linear Elastic Material** > **Inelastic Strain Rate**
Layered Shell > **Hyperelastic Material** > **Inelastic Strain Rate**
Membrane > **Linear Elastic Material** > **Inelastic Strain Rate**
Membrane > **Linear Elastic Material, Layered** > **Inelastic Strain Rate**
Membrane > **Hyperelastic Material** > **Inelastic Strain Rate**
Membrane > **Nonlinear Elastic Material** > **Inelastic Strain Rate**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Hyperelastic Material**, **Linear Elastic Material, Layered**, or **Hyperelastic Material, Layered** node selected in the model builder tree:

Attributes > **Inelastic Strain Rate**

Damage

Use the **Damage** subnode to model damage and cracking in brittle materials according to various criteria. It is available with **Linear Elastic Material** in the Solid Mechanics and Layered Shell interfaces, and with **Linear Elastic Material, Layered** in the Shell interface. **Phase-field damage** is also available for **Hyperelastic Material** in the Solid Mechanics interface.

The **Damage** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/). **Damage** is available for 3D, 2D, and 2D axisymmetry.

When a **Damage** node is present, a result template named **Damage** is added. It shows the distribution of the damage variable, ranging from 0 to 1.

SHELL PROPERTIES

This section is present when **Damage** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Damage node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Damage node in the Shell and Plate chapter.

DAMAGE

Select the type of **Damage model** — **Scalar damage**, **Mazars damage for concrete**, or **Phase-field damage**. Then follow the instructions below.

Scalar Damage

Select the type of **Equivalent strain** — **Rankine, stress**; **Rankine, strain**; **Smooth Rankine, stress**; **Smooth Rankine, strain**; **Norm of elastic strain tensor**; or **User defined**. See Strain-based Damage Models for details.

The **Activate damage in compression** checkbox is available for the **Rankine, stress**; **Rankine, strain**; **Smooth Rankine, stress**; **Smooth Rankine, strain**; or **User defined** equivalent strain definitions, and it is not selected by default. When selected, the damage evolution law is applied on the total undamaged stress tensor.

Select the type of **Damage evolution** — **Linear strain softening**, **Exponential strain softening**, **Polynomial strain softening**, **Multilinear strain softening**, or **User defined**. See Damage Evolution for details.

- For **Linear strain softening**, **Polynomial strain softening**, **Multilinear strain softening**, or **Exponential strain softening** enter the **Tensile strength** σ_{ts}, the default is to take the value **From material**. Change to **User defined** to enter other value or expression.
- For **Linear strain softening** or **Exponential strain softening** select the type of **Strain softening input** and enter the **Fracture energy per area** G_f, the **Fracture energy per**

volume g_f, or the **Strain softening parameter** ε_f accordingly. The default is to take the value **From material**. Change to **User defined** to enter other value or expression. The available options depend on the spatial regularization method selected.

- For **Multilinear strain softening**, enter the **Shape factor** λ. The default value is 0.5.
- For **User defined**, enter an expression for the **Damage evolution** function $d(\kappa)$.

Select the type of **Spatial regularization method** — **None**, **Crack band**, or **Implicit gradient**. See Spatial Regularization for details.

- For the **Crack band** method select the type of **Crack band calculation** — **Element volume/area** or **Element size**. Note that when the Crack band method is selected, only the **Fracture energy per area** is available as **Strain softening input**.
- For the **Implicit gradient** method enter the **Internal length scale** l_{int}. If the **Fracture energy per area** was selected as **Strain softening input**, enter also the **Characteristic size of the damage dissipation zone** h_{dmg}. The **Implicit gradient** method is available in the Solid Mechanics and Layered Shell interfaces.

Select the type of **Viscous regularization method** — **None** or **Delayed Damage**.

For the **Delayed damage** method, enter the **Characteristic time** τ. The Delayed damage method is intended for time-dependent studies, and adds no contributions for other study types. See Viscous Regularization for details.

Mazars Damage for Concrete

Select the type of **Equivalent strain** — **Mazars**, **Modified Mazars**, or **User defined**. See Mazars Damage for Concrete for details.

Enter the **Shear exponent** β, the default is set to 1.06.

Select the **Tensile damage evolution** — **Linear strain softening**, **Exponential strain softening**, **Mazars damage evolution function**, or **User defined**.

- For **Linear strain softening** or **Exponential strain softening** enter the **Tensile strength** σ_{ts}, the default is to take the value **From material**. Also select the type of **Tensile strain softening** and enter the **Fracture energy per area** G_{ft}, the **Fracture energy per volume** g_{ft}, or the **Strain softening parameter** ε_{ft} accordingly. The default is to take the value **From material**. Change to **User defined** to enter other value or expression.
- For **Mazars damage evolution function**, enter the **Tensile strain threshold** ε_{0t}, and the **Tensile damage evolution parameters** A_t and B_t.
- For **User defined**, enter an expression for the **Tensile damage evolution** function $d_t(\kappa)$.

Select the **Compressive damage evolution** — **Mazars damage evolution function** or **User defined**.

- For **Mazars damage evolution function**, enter the **Compressive strain threshold** ε_{0c}, and the **Compressive damage evolution parameters** A_c and B_c.
- For **User defined**, enter an expression for the **Compressive damage evolution** function $d_c(\kappa)$.

Select the type of **Spatial regularization method** — **None**, **Crack band**, or **Implicit gradient**. See Spatial Regularization for details.

- For the **Crack band** method select the type of **Crack band calculation** — **Element volume/area** or **Element size**. Note that when the Crack band method is selected, only the **Fracture energy per area** is available as **Strain softening input**.
- For the **Implicit gradient** method enter the **Length scale** l_{int}. If the **Fracture energy per area** was selected as **Strain softening input**, enter also the **Characteristic size of the damage dissipation zone** h_{dmg}. The **Implicit gradient** method is available in the Solid Mechanics and Layered Shell interfaces.

Select the type of **Viscous regularization method** — **None** or **Delayed Damage**.

For the **Delayed damage** method, enter the **Characteristic time** τ. The Delayed damage method is intended for time-dependent studies, and adds no contributions for other study types. See Viscous Regularization for details.

Phase-Field Damage

The **Phase-field damage** model is available with the **Linear Elastic Material** or the **Hyperelastic Material** in the Solid Mechanics interface. See Phase-Field Damage Models for details.

Select the type of **Crack driving force** — **Elastic strain energy density**, **Total strain energy density**, **Principal stress criterion**, or **User defined**. See Crack Driving Force for details.

- For **Elastic strain energy density** and **Total strain energy density**, enter a value for the **Critical energy release rate** G_c, and the **Strain energy threshold** G_{c0}. The default for the **Critical energy release rate** is to take the value **From material**. Change to **User defined** to enter other value or expression.
- For **Principal stress criterion**, enter a value for the **Critical fracture stress** σ_c, and the **Post-peak slope parameter** ξ. The default for the **Critical fracture stress** is to take the value **From material**. Change to **User defined** to enter other value or expression.
- For **User defined**, enter an expression for the dimensionless crack driving force.

For all options, enter a value for the **Length scale** l_{int}.

Select the **Damage evolution** function to use — **Power law**; **Cubic, Borden**; or **User defined**. See Damage Evolution for Phase-Field Damage for details.

- For **Power law**, enter a value for the **Exponent** m. The default is to use a quadratic function so that $m = 2$.
- For **Cubic, Borden**, enter a value for the **Model parameter** s. This option is available when **Crack driving force** is set to **Elastic strain energy density** or **Total strain energy density**.
- For **User defined**, enter an expression for the **Damage evolution** function $d(\phi)$.

Select an option for **Exclude compressive energy** — **Volumetric only**; **Spectral decomposition, stress**; **Spectral decomposition, strain**; or **No split**. For **Hyperelastic Material**, only some options are available from the list. See Strain Energy Split for details.

The **Viscous regularization** checkbox is not selected by default. When selected, a viscous term is added to the evolution of the crack phase field in time-dependent studies. Enter a value for the **Characteristic time** τ. See Viscous Regularization for details.

DISCRETIZATION

This section is available with the **Implicit gradient** regularization method and the **Phase-field damage** model. Select the shape function for the **Crack phase field** ϕ — **Automatic**, **Linear**, **Quadratic Lagrange**, **Quadratic serendipity**, **Cubic Lagrange**, **Cubic serendipity**, **Quartic Lagrange**, **Quartic serendipity**, or **Quintic Lagrange**. The options available depends on the chosen order of the displacement field.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

ADVANCED

Enter the **Maximum damage**. The default value is 0.99999.

⚠	When a Mixed Formulation is selected in the parent **Linear Elastic Material**, the damage model can give spurious results. The **Implicit gradient** regularization method is more stable in this respect, and it is recommended when using the mixed formulation.

See also

- Modeling Damage in the *Structural Mechanics Modeling* chapter.
- Damage Models in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Damage**
Solid Mechanics > **Hyperelastic Material** > **Damage**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Damage**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Damage**
Shell > **Linear Elastic Material, Layered** > **Damage**
Layered Shell > **Linear Elastic Material** > **Damage**

Ribbon

Physics tab with **Linear Elastic Material**, **Hyperelastic Material**, or **Linear Elastic Material, Layered**, node selected in the model tree:

Attributes > **Damage**

Activation

Use the **Activation** subnode to activate or deactivate parts of a domain according to an **Activation expression** that you define. The **Activation** subnode uses a scale factor that reduces the elastic stiffness and density of the material, to emulate that material is not present. When the **Activation expression** is satisfied, the material is activated in a state of zero stress.

When the material is not active, the following is done to emulate that the material is not present:

- The elasticity matrix is multiplied by the **Activation scale factor**.
- The density is multiplied by the square of the **Activation scale factor**.
- If viscoelasticity is present, shear and bulk moduli, as well as relaxation times, are scaled using the **Activation scale factor**.

When the material is activated, the following is done to ensure that the activation is stress free:

- At the instant of activation, that is when the **Activation expression** is first satisfied, the elastic strains corresponding to the previous time or parameter step are removed from the total strains.
- Initial stresses are always removed, regardless of the **Activation expression**.

External stresses are never removed, regardless of the **Activation expression**.

The **Activation** subnode can be used in combination with **Linear Elastic Material**, and in the Wire interface with **Elastic Wire**.

ACTIVATION

Enter the **Activation expression**. This is the expression that will determine whether material is active or inactive.

Enter the **Activation scale factor**. The default value is 10^{-5}.

The **Activation** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/)

See also Activating and Deactivating Material in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Activation**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Activation**
Layered Shell > **Linear Elastic Material** > **Activation**
Membrane > **Linear Elastic Material** > **Activation**
Truss > **Linear Elastic Material** > **Activation**
Truss, Explicit Dynamics > **Linear Elastic Material** > **Activation**
Wire > **Elastic Wire** > **Activation**

Ribbon

Physics tab with **Linear Elastic Material** or **Elastic Wire** node selected in the model tree:

Attributes > **Activation**

Safety

Use the **Safety** subnode to set up variables that can be used to check the risk of failure according to various criteria. It can be used in combination with **Linear Elastic Material**, **Nonlinear Elastic Material**, and **Linear Elastic Material, Layered**. Four different variables describing the failure risk are defined, as described in Table 4-2.

You can add any number of **Safety** nodes to a single material model. The contents of this feature do not affect the analysis results as such, so you can add **Safety** nodes after having performed an analysis, and just do an **Update Solution** ( ) in order to access to the new variables for result evaluation.

TABLE 4-2: VARIABLES FOR SAFETY FACTOR EVALUATION.

VARIABLE	DESCRIPTION	CRITERION FULFILLED	CRITERION VIOLATED
Failure index, FI	For a linear criterion, this is the ratio between the computed value and the given limit.	FI<1	FI>1
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=1
Safety factor, SF	For a linear criterion, this is 1/FI.	SF>1	SF<1
Margin of safety, MoS	SF-1	MoS>0	MoS<0

For orthotropic and anisotropic failure criteria, the directions are given by the coordinate system selection in the parent node.

When you add a **Safety** node in one of the Shell, Layered Shell, or Membrane interfaces, a default plot with the failure index is generated. Such plots are placed in a group named **Failure Indices**. The label of these plots is derived from the label of the corresponding **Safety** node.

The **Safety** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

SHELL PROPERTIES

This section is present when **Safety** is used as a subnode to:

- **Linear Elastic Material** in the Layered Shell interface. See the documentation for the Safety node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Safety node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Safety node in the Membrane chapter.

FAILURE MODEL

Select a **Failure Criterion**. The available choices depend on the physics interface, as indicated in Table 4-3.

TABLE 4-3: AVAILABLE FAILURE CRITERIA BY PHYSICS INTERFACE.

CRITERION	SOLID MECHANICS	SHELL, PLATE	LAYERED SHELL	MEMBRANE	BEAM, PIPE MECHANICS	TRUSS
von Mises	X	X	X	X	X	X
Tresca	X	X	X	X	X	X
Rankine	X	X	X	X	X	X
Saint-Venant	X	X	X	X	X	X
Mohr–Coulomb	X	–	–	–	–	–
Drucker–Prager	X	–	–	–	–	–
Griffith	X	X	X	X	X	–
Bresler–Pister	X	X	–	–	–	–
Willam–Warnke	X	X	–	–	–	–
Ottosen	X	X	–	–	–	–
Jenkins	X	X	X	X	–	–
Waddoups	X	X	X	X	–	–
Azzi–Tsai–Hill	Plane stress	X	X	X	–	–
Norris	Plane stress	X	X	X	–	–
Tsai–Hill	X	X	X	X	–	–
Hoffman	X	X	X	X	–	–

TABLE 4-3: AVAILABLE FAILURE CRITERIA BY PHYSICS INTERFACE.

CRITERION	SOLID MECHANICS	SHELL, PLATE	LAYERED SHELL	MEMBRANE	BEAM, PIPE MECHANICS	TRUSS
Tsai–Wu Orthotropic	X	X	X	X	–	–
Zinoviev[1]	–	X	X	–	–	–
Hashin–Rotem[1]	–	X	X	–	–	–
Hashin[1]	–	X	X	–	–	–
Puck[1]	–	X	X	–	–	–
LaRC03[1]	–	X	X	–	–	–
Tsai–Wu Anisotropic	X	X	X	X	–	–
User defined	X	X	X	X	X	X

1) Requires the Composite Materials Module

- When **Failure Criterion** is **von Mises**, enter **Tensile strength** σ_{ts}.
- When **Failure Criterion** is **Tresca**, enter **Tensile strength** σ_{ts}.
- When **Failure Criterion** is **Rankine**, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs}.
- When **Failure Criterion** is **Saint-Venant**, enter **Ultimate tensile strain** ε_{ts} and **Ultimate compressive strain** ε_{cs}.
- When **Failure Criterion** is **Mohr–Coulomb**, select **Material parameters** — **Cohesion and angle of friction** or **Tensile and compressive strengths** to determine the type of input data.

  When **Cohesion and angle of friction** is used, enter **Cohesion** c and **Angle of internal friction** ϕ.

  When **Tensile and compressive strengths** is used, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs}.

  In either case, you can select **Include elliptic cap** to limit the allowed compressive stress. When selected, enter the **Elliptic cap parameters** p_a and p_b.

- When **Failure Criterion** is **Drucker–Prager**, select **Material parameters** — **Drucker–Prager parameters**, **Tensile and compressive strengths**, or **Mohr–Coulomb parameters** to determine the type of input data.

  When **Drucker–Prager parameters** is used, enter **Drucker–Prager alpha coefficient** α and **Drucker–Prager k coefficient** k.

  When **Tensile and compressive strengths** is used, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs}.

  When **Mohr–Coulomb parameters** is used, enter **Cohesion** c and **Angle of internal friction** ϕ.

  In either case, you can select **Include elliptic cap** to limit the allowed compressive stress. When selected, enter the **Elliptic cap parameters** p_a and p_b.
- When **Failure Criterion** is **Griffith**, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs}.
- When **Failure Criterion** is **Bresler–Pister**, enter **Tensile strength** σ_{ts}, **Compressive strength** σ_{cs}, and **Biaxial compressive strength** σ_{bc}.
- When **Failure Criterion** is **Willam–Warnke**, enter **Tensile strength** σ_{ts}, **Compressive strength** σ_{cs}, and **Biaxial compressive strength** σ_{bc}.
- When **Failure Criterion** is **Ottosen**, enter the **Compressive strength** σ_{cs}, **Ottosen parameters** a and b, the **Size factor** k_1, and the **Shape factor** k_2.
- When **Failure Criterion** is **Jenkins**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Waddoups**, enter **Ultimate tensile strains** ε_{ts}, **Ultimate compressive strains** ε_{cs}, and **Ultimate shear strains** γ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Azzi–Tsai–Hill**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Norris**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Tsai–Hill**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy. Select the **Use plane stress formulation** checkbox to assume plane stress conditions, see Tsai–Hill Criterion.

- When **Failure Criterion** is **Hoffman**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Tsai–Wu Orthotropic**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy. Select the **Use plane stress formulation** checkbox to assume plane stress conditions, see Orthotropic Tsai–Wu Criterion.
- When **Failure Criterion** is **Zinoviev**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hashin–Rotem**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hashin**, enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy. Select the **Use plane stress formulation** checkbox to assume plane stress conditions, see Hashin Criterion.
- When **Failure Criterion** is **Puck**:
  - Enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy.
  - Enter the **Fiber failure** data: **Ultimate tensile strain in longitudinal direction**, ε_{ts1}, and **Ultimate compressive strain in longitudinal direction**, ε_{cs1}. Also, enter the fiber material properties **Young's modulus of fiber in longitudinal direction**, E_{f1}, and **In-plane Poisson's ratio of fiber**, ν_{f12}. Enter a **Mean stress magnification factor**, m_f. The default value is 1.3, a value commonly assumed for GFRP. For CFRP, the value 1.1 has been suggested.
  - Enter the **Interfiber failure** data: **Linear degradation stress**, σ_{1D}. Also, enter the **Slope of in-plane fracture envelope, tension**, p_{tl}, and the **Slope of in-plane fracture envelope, compression**, p_{cl}. The default values are 0.3 and 0.25, respectively. These values are common for GFRP. For CFRP, the values 0.35 and 0.3 have been suggested.

- When **Failure Criterion** is **LaRC03**:
  - Enter **Tensile strengths** σ_{ts}, **Compressive strengths** σ_{cs}, and **Shear strengths** σ_{ss}. All entries have three components, related to the principal axes of orthotropy. Enter the **Ultimate tensile strain in longitudinal direction**, ε_{tsl}.
  - Enter the **In situ transverse tensile strength**, σ^{i}_{ts2}, and the **In situ in-plane shear strength**, σ^{i}_{ss12}. For both parameters, the default is to compute the value as indicated by the selections **From transverse tensile strength** and **From in-plane shear strength** respectively. You can also take the values **From material**, or select **User defined** to enter other expressions. See LARC-03 Criterion.
  - When the parent Linear Elastic Material is anisotropic, enter the **Young's modulus in longitudinal direction**, E_1, the **Young's modulus in transverse direction**, E_2, the **In-plane Poisson's ratio**, ν_{12}, and the **In-plane shear modulus**, G_{12}.
  - Enter the **Fracture plane angle under uniaxial transverse compression**, α_0, and the **Fracture plane search resolution**, $\Delta\alpha$. The default values are $53°$ and $3°$ respectively. Under combined loading, the fracture plane angle α will differ from α_0, and a numerical search for the critical angle is performed in the range $0 < \alpha < \alpha_0$ with a step of $\Delta\alpha$ degrees.
- When **Failure Criterion** is **Tsai–Wu Anisotropic**, enter the **Second rank tensor, Voigt notation F**, and the **Fourth rank tensor F**. Enter the components of the tensors with respect to the directions of the coordinate system in the parent node, see Anisotropic Tsai–Wu Criterion.
- When **Failure Criterion** is **User defined**, enter two expressions describing the **Failure criterion** $g(S)$ and the **Safety factor** $s_f(S)$. As an example, to replicate the **von Mises Isotropic** criterion with a tensile strength of 350 MPa, define $g(S)$ as `solid.mises/350[MPa]-1` and $s_f(S)$ as `350[MPa]/(solid.mises+eps)`.

For all input fields, the default is to take the value **From material**. Change to **User defined** to enter other values or expressions.

- For a detailed description of the various criteria and how to combine them, see Safety Factor Evaluation in the *Structural Mechanics Theory* chapter.
- Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Variables** > **Safety**
Solid Mechanics > **Nonlinear Elastic Material** > **Variables** > **Safety**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Variables** > **Safety**
Shell > **Linear Elastic Material** > **Variables** > **Safety**
Shell > **Linear Elastic Material, Layered** > **Variables** > **Safety**
Layered Shell > **Linear Elastic Material** > **Variables** > **Safety**
Plate > **Linear Elastic Material** > **Variables** > **Safety**
Membrane > **Linear Elastic Material, Layered** > **Variables** > **Safety**
Membrane > **Linear Elastic Material** > **Variables** > **Safety**
Membrane > **Nonlinear Elastic Material** > **Variables** > **Safety**
Beam > **Linear Elastic Material** > **Variables** > **Safety**
Truss > **Linear Elastic Material** > **Variables** > **Safety**
Truss, Explicit Dynamics > **Linear Elastic Material** > **Variables** > **Safety**
Pipe Mechanics > **Fluid and Pipe Materials** > **Variables** > **Safety**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Fluid and Pipe Materials** node selected in the model builder tree:

Attributes > **Variables** > **Safety**

Damping

Use the **Damping** subnode to add several types of damping to the material model. Damping can be used in Time Dependent, Eigenfrequency, and Frequency Domain studies; for other study types the settings in the **Damping** subnode are ignored.

The following types of damping are available:

- Rayleigh Damping
- Isotropic Loss Factor
- Anisotropic Loss Factor
- Viscous Damping
- Loss Factor

- Maximum Loss Factor
- Wave Attenuation

The option **Maximum loss factor** is not available in the Solid Mechanics, Explicit Dynamics interface.

The available damping models differs between various COMSOL products (see www.comsol.com/products/specifications/).

For a detailed description of the various damping models, see Mechanical Damping and Losses in the *Structural Mechanics Modeling* chapter.

The applicability of the different damping models depends on the physics interface, material model, and study type.

SHELL PROPERTIES

This section is only present when **Damping** is used as a subnode to:

- A material model in the Layered Shell interface. See the documentation for the Damping node in the Layered Shell chapter.
- **Linear Elastic Material, Layered** in the Shell interface. See the documentation for the Damping node in the Shell and Plate chapter.
- **Linear Elastic Material, Layered** in the Membrane interface. See the documentation for the Damping node in the Membrane chapter.

DAMPING SETTINGS

Select a **Damping type**, and enter the settings accordingly.

Rayleigh Damping

This choice can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. In this model, the damping ratio ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{dM} m + \beta_{dK} k$$

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* α_{dM} and the *stiffness damping parameter* β_{dK}.

Select **Input parameters** — **Alpha and beta** — to enter the damping parameters explicitly, or **Damping ratios** to derive the damping parameters from the damping ratio at two frequencies.

When **Alpha and beta** is selected, enter values or expressions for the **Mass damping parameter** α_{dM} and the **Stiffness damping parameter** β_{dK}.

When **Damping ratios** is selected, enter two pairs of frequencies, f_1 and f_2, and the corresponding damping ratios ζ_1 and ζ_2 at these frequencies. The Rayleigh damping parameters are then computed as

$$\alpha_{dM} = 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{dK} = \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi(f_2^2 - f_1^2)}$$

In order to visualize the damping ratio as a function of frequency, click **Damping Ratio Preview** ( ).

Rayleigh Damping in the **Beam** interfaces is ignored for geometrically nonlinear analysis.

Isotropic Loss Factor

This choice is effective only in Eigenfrequency and Frequency Domain study. The isotropic loss factor damping is described by the single isotropic loss factor η_s, which acts on all entries in the elastic constitutive matrix. It can be used for isotropic, orthotropic, and anisotropic materials.

When **Isotropic loss factor** is selected, use the **Isotropic structural loss factor** list to select the way to enter η_s. The default is to take the value **From material**. For **User defined**, enter another value or expression.

Anisotropic Loss Factor

This choice is effective only in Eigenfrequency and Frequency Domain study. An elastic material is in general described by a symmetric 6-by-6 elasticity matrix $\boldsymbol{D}$. The loss can be isotropic or anisotropic, and is described by either the isotropic loss factor

η_s or by a symmetric anisotropic 6-by-6 loss factor matrix η_D or η_{DVo}. The orientations are the same as in the parent node.

When **Anisotropic loss factor** is selected, use the **Loss factor for elasticity matrix D** list to select the way to enter η_D or η_{DVo}. The default is to take the values **From material**. For **User defined** enter the components of η_D or η_{DVo} in the upper-triangular part of a symmetric 6-by-6 matrix.

The values for the loss factors are ordered in two different ways, consistent with the selection of either **Standard (XX, YY, ZZ, XY, YZ, XZ)** or **Voigt (XX, YY, ZZ, YZ, XZ, XY)** notation in the corresponding Linear Elastic Model. If the values are taken from the material, these loss factors are found in the **Anisotropic** or **Anisotropic, Voigt notation** property group for the material. For an isotropic material, the anisotropic loss factor is always given as η_D using the standard notation.

Viscous Damping

This choice can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. With viscous damping, the material will get additional stresses proportional to the strain rate. Enter **Bulk viscosity** η_b and **Shear viscosity** η_v to model damping caused by volume change and deformation, respectively.

See also Viscous Damping in the *Structural Mechanics Theory* chapter.

Maximum Loss Factor

This damping model can be used in Eigenfrequency, Frequency Domain, and Time Dependent studies. Enter the **Maximum loss factor** η_{max} together with the **Reference frequency** f_{ref}, at which the maximum loss factor occurs. The model approximates the maximum loss factor around the provided reference frequency.

See also Maximum Loss Factor in the *Structural Mechanics Theory* chapter.

Wave Attenuation

This damping model is only available when **Isotropic** is selected as the **Material symmetry** in the parent Linear Elastic Material feature. It can be used in

Eigenfrequency, Frequency Domain, and Time Dependent study. Enter the elastic wave spatial **Attenuation coefficient** for the pressure waves (p-subscript) and shear waves (s-subscript) together with the **Reference frequency** $f_{\mathrm{p,ref}}$ and $f_{\mathrm{s,ref}}$ at which the respective coefficient was measured. You can also select the **Attenuation unit** for the **Attenuation coefficient** inputs. The available options are: decibel (dB) per wavelength, neper (Np) per wavelength, decibel per unit length, and neper per unit length. The damping model is similar to Viscous Damping, for which the software will use the effective bulk and shear viscosity computed automatically based on the attenuation inputs.

See also Wave Attenuation in the *Structural Mechanics Theory* chapter.

Loss Factor

This option is available when working with the Section Stiffness material model in the Shell interface.

The loss factors are directly acting on the components of the different matrices. Each component in the given damping matrix acts on the corresponding entry in the stiffness matrix.

Enter the **Loss factor for stiffness matrix DA**, η_{DA}; **Loss factor for stiffness matrix DB**, η_{DB}; **Loss factor for stiffness matrix DD**, η_{DD}; and **Loss factor for stiffness matrix DAs**, η_{DAs}.

The default for all section properties is to take the values **From material**. Any one of the loss matrices can also be **User defined**. In that case, selecting **Isotropic** input is identical to selecting **Diagonal** input and entering the same value in all three diagonal components. In most cases, the **Symmetric** input option is the most relevant, since that

is the only one in which a loss factor can be assigned to all elements in the section stiffness matrices.

- The time-stepping algorithms also add numerical damping, which is independent of any explicit damping added. For the generalized alpha time-stepping algorithm you can control the amount of numerical damping.
- In a study using mode superposition, you can add damping to the individual eigenmodes.

For an example of Damping, see *Heat Generation in a Vibrating Structure*: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/vibrating_beam**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Damping**
Solid Mechanics > **Nonlinear Elastic Material** > **Damping**
Solid Mechanics > **Hyperelastic Material** > **Damping**
Solid Mechanics > **Elastoplastic Soil Material** > **Damping**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Damping**
Solid Mechanics, Explicit Dynamics > **Hyperelastic Material** > **Damping**
Shell > **Linear Elastic Material** > **Damping**
Shell > **Linear Elastic Material, Layered** > **Damping**
Shell > **Layered Linear Hyperelastic Material** > **Damping**
Shell > **Section Stiffness** > **Damping**
Layered Shell > **Linear Elastic Material** > **Damping**
Layered Shell > **Hyperelastic Material** > **Damping**
Membrane > **Linear Elastic Material** > **Damping**
Membrane > **Linear Elastic Material, Layered** > **Damping**
Membrane > **Nonlinear Elastic Material** > **Damping**
Membrane > **Hyperelastic Material** > **Damping**
Plate > **Linear Elastic Material** > **Damping**
Plate > **Section Stiffness** > **Damping**
Beam > **Linear Elastic Material** > **Damping**
Beam > **Section Stiffness** > **Damping**

Pipe Mechanics > **Fluid and Pipe Properties** > **Damping**
Truss > **Linear Elastic Material** > **Damping**
Truss, Explicit Dynamics > **Linear Elastic Material** > **Damping**
Wire > **Elastic Wire** > **Damping**
Multibody Dynamics > **Linear Elastic Material** > **Damping**
Solid Rotor > **Linear Elastic Material** > **Damping**
Solid Rotor, Fixed Frame > **Linear Elastic Material** > **Damping**

Ribbon
Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material**, **Layered Linear Hyperelastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Section Stiffness**, **Fluid and Pipe Properties**, or **Elastic Wire** node selected in the model tree:

Attributes > **Damping**

Mechanical Damping

The **Mechanical Damping** subnode allows you to model mechanical losses in the Piezoelectric Material and Piezomagnetic Material either using loss factor material data for the stiffness, or in the form of Rayleigh proportional damping.

The **Mechanical Damping** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

DAMPING SETTINGS

Select a **Damping type** — **Loss factor for cE**, **Loss factor for sE**, **Loss factor for cH**, **Loss factor for sH**, **Isotropic loss factor**, **Rayleigh damping**, or **Maximum loss factor**.

Loss Factor for cE (Piezoelectric Material)
This choice is effective only in Eigenfrequency and Frequency Domain study. By default, the **Loss factor for elasticity matrix cE** η_{cE} uses values **From material**. For **User defined** enter other values or expressions in the matrix.

Loss Factor for sE (Piezoelectric Material)
This choice is only available if the **Constitutive relation** in the parent node is **Strain–charge form**, and it is effective only in Eigenfrequency and Frequency Domain study. By default, the **Loss factor for compliance matrix sE** η_{sE} uses values **From material**. For **User defined** enter other values or expressions in the matrix.

Loss Factor for cH (Piezomagnetic Material)

This choice is effective only in Eigenfrequency and Frequency Domain study. By default, the **Loss factor for elasticity matrix cH** η_{cH} uses values **From material**. For **User defined** enter other values or expressions in the matrix.

Loss Factor for sH (Piezoelectric Material)

This choice is only available if the **Constitutive relation** in the parent node is **Strain–magnetization form**, and it is effective only in Eigenfrequency and Frequency Domain study. By default, the **Loss factor for compliance matrix sH** η_{sH} uses values **From material**. For **User defined** enter other values or expressions in the matrix.

Isotropic Loss Factor

This choice is effective only in Eigenfrequency and Frequency Domain study. By default, the **Isotropic structural loss factor** η_{s} uses values **From material**. For **User defined** enter another value or expression.

Rayleigh Damping

This damping model can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. Enter the **Mass damping parameter** $\alpha_{\mathrm{d}M}$ and the **Stiffness damping parameter** $\beta_{\mathrm{d}K}$.

The damping ratio ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{\mathrm{d}M} m + \beta_{\mathrm{d}K} k$$

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* $\alpha_{\mathrm{d}M}$ and the *stiffness damping parameter* $\beta_{\mathrm{d}K}$. Note that the beta-damping is applied only to the mechanical part of the problem.

In order to visualize the damping ratio as a function of frequency, click **Damping Ratio Preview** ( ).

To include the Rayleigh damping effect for the piezoelectric coupling terms, add a Coupling Loss subnode.

Maximum Loss Factor

This choice can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. Enter the **Maximum loss factor** η_{max} together with the **Reference frequency** f_{ref}, at which the maximum occurs.

	See also Piezoelectric Losses.
	See also Maximum Loss Factor in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Piezoelectric Material** > **Mechanical Damping**
Solid Mechanics > **Piezomagnetic Material** > **Mechanical Damping**
Layered Shell > **Piezoelectric Material** > **Mechanical Damping**

Ribbon

Physics tab with **Piezoelectric Material** or **Piezomagnetic Material** node selected in the model tree:

Attributes > **Mechanical Damping**

Coupling Loss

The **Coupling Loss** subnode allows you to model losses in the piezoelectric coupling in a Piezoelectric Material, either by using the loss factor material data for the coupling matrix or as Rayleigh proportional damping.

The **Coupling Loss** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COUPLING LOSS SETTINGS

Select a **Coupling loss** — **Loss factor for e** or **Rayleigh damping**.

- For **Loss factor for e** select a **Loss factor for coupling matrix e** from the list. Select **User defined** to enter values or expressions for η_e in a 3-by-6 matrix. This choice is effective only in Eigenfrequency and Frequency Domain study.
- For **Rayleigh damping** enter a **Stiffness damping parameter** β_{dC}. This choice can be used in Eigenfrequency Frequency Domain, and Time Dependent study.

See also Piezoelectric Losses.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Piezoelectric Material** > **Coupling Loss**
Layered Shell > **Piezoelectric Material** > **Coupling Loss**

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:

Attributes > **Coupling Loss**

Dielectric Loss

The **Dielectric Loss** subnode allows you to model electrical losses in the Piezoelectric Material. The losses can be prescribed either by using a loss factor for the electric permittivity, or in the form of dielectric dispersion.

The **Dielectric Loss** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

DIELECTRIC LOSS SETTINGS

From the **Dielectric loss** list, select one of the following options:

- **Loss factor for** ε_S. This choice is effective only in Eigenfrequency and Frequency Domain studies. You can select the input type for **Loss factor for electric permittivity** ε_S to be either **From material** to use the value from the material, or from **User defined** to enter values or expressions for the loss factor in the associated fields. Select **Symmetric** to enter the components of $\eta_{\varepsilon S}$ in the upper-triangular part of a

symmetric 3-by-3 matrix, select **Isotropic** to enter a single scalar loss factor, or select **Diagonal**. The default values are 0.

- **Loss factor for** εT. This choice is only available if the **Constitutive relation** in the parent node is **Strain–charge form**, and it is effective only in Eigenfrequency and Frequency Domain study. You can select the input type for **Loss factor for electric permittivity** εT to be either **From material** to use the value from the material, or from **User defined** to enter values or expressions for the loss factor in the associated fields. Select **Symmetric** to enter the components of $\eta_{\varepsilon T}$ in the upper-triangular part of a symmetric 3-by-3 matrix, select **Isotropic** to enter a single scalar loss factor, or select **Diagonal**. The default values are 0.
- **Dispersion**. This choice can be used in Eigenfrequency Frequency Domain, and Time Dependent study. Enter the **Relaxation time** τ_d, and the **Relative permittivity contribution** $\Delta\varepsilon_{rS}$ in the associated fields. For the latter, you can select **Isotropic**, **Diagonal**, or **Symmetric** matrix input options. You can also specify how the relative permittivity input ε_{rS} on the parent node should be interpreted by selecting the **Static response** (the default value is **Low frequency limit**).
- **Complex permittivity**. This choice can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. You enter the **Relative permittivity (real part)** ε' and **Relative permittivity (imaginary part)** ε'', which can be either **Isotropic** or **Diagonal** matrices. You also enter the **Reference frequency** f_{ref} at which the permittivity values have been measured. You can also specify how the relative permittivity input ε_{rS} on the parent node should be interpreted by selecting the **Static response** (the default value is **Low frequency limit**). You can also specify how the relative permittivity input ε_{rS} on the parent node should be interpreted by selecting the **Static response** (the default value is **Low frequency limit**). In Time Dependent study, the software will use the dispersion model, for which the effective relaxation time and relative permittivity increment are computed automatically based on the node input parameters.
- **Maximum loss tangent**. This choice can be used in Eigenfrequency, Frequency Domain, and Time Dependent study. You enter the **Maximum loss tangent** η_{max} together with the **Reference frequency** f_{ref} at which the maximum occurs. Both inputs can be either **Isotropic** or **Diagonal** matrices. In the latter case, different frequency can be used for the corresponding components of the loss tangent. You can also specify how the relative permittivity input ε_{rS} on the parent node should be interpreted by selecting the **Static response** (the default value is **Low frequency limit**). In Time Dependent study, the software will use the dispersion model, for which the

effective relaxation time and relative permittivity increment are computed automatically based on the node input parameters.

	See also Piezoelectric Losses.

	If your license includes either the AC/DC Module or MEMS Module, more options for modeling dielectric dispersion can be found in Dispersion section in the *AC/DC Module User's Guide*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Piezoelectric Material** > **Dielectric Loss**
Layered Shell > **Piezoelectric Material** > **Dielectric Loss**

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:

Attributes > **Dielectric Loss**

Conduction Loss (Time-Harmonic)

The **Conduction Loss (Time-Harmonic)** subnode allows you to model possible conductive losses in a Piezoelectric Material. The effect is only active in an eigenfrequency or frequency-domain study.

The **Conduction Loss (Time-Harmonic)** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

CONDUCTION CURRENT

By default, the **Electric conductivity** σ_e for the medium is defined **From material**. You can also select **User defined** or **Linearized resistivity**.

- For **User defined** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** depending on the characteristics of the electric conductivity, and then enter values or expressions for the **Electric conductivity** σ_e in the field or matrix.
- For **Linearized resistivity** the default **Reference temperature** T_0, and **Resistivity temperature coefficient** α_r, and **Reference resistivity** ρ_0 are taken **From material**, which

means that the values are taken from the domain (or boundary) material. T is the current temperature, which can be a value that is specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field appears in the **Model Inputs** section.

See also Piezoelectric Losses.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Piezoelectric Material** > **Conduction Loss (Time-Harmonic)**
Electrostatics > **Charge Conservation** > **Conduction Loss (Time-Harmonic)**

Ribbon

Physics tab with **Piezoelectric Material** or **Charge Conservation** node selected in the model tree:

Attributes > **Conduction Loss (Time-Harmonic)**

External Stress–Strain Relation

The **External Stress–Strain Relation** is a special type of material model where the computation of stress is delegated to external code which has been compiled into a shared library. External libraries must first be imported into an **External Material** feature under **Global Definitions** > **Materials**.

See also External Material and Working with External Materials in the *COMSOL Multiphysics Reference Manual*.

The **External Stress–Strain Relation** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. The coordinate system must have

orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

MATERIAL

Select an **External material** from the list of compatible external materials added under **Global Definitions** > **Materials**. For a material to be compatible with this **External Material** model node, its **Interface type** must be set to a type whose required input quantities are all defined by this external material. Allowed required inputs include strain, the deformation gradient, as well as all standard model inputs.

The stress and strain measures that are used will depend on the kinematics. For example, in a total Lagrangian formulation, strains will be interpreted as Green–Lagrange strains, and output stresses are expected to be second Piola–Kirchhoff stresses.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

TRANSVERSE SHEAR STRAINS

This section is only present in the Membrane interface and in 2D Plane Stress in the Solid Mechanics interface.

As a default, it is assumed that the transverse shear strains are zero. For a state of plane stress, this is true for an isotropic material, and for nonisotropic materials where the normal to the surface is a principal direction. If this is not the case, select the **Transverse shear strains** checkbox to store state variables also for the two transverse shear strains.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points used for the weak contribution in the material. Reduced integration give faster computations at the element level and help mitigate locking issues. However, reduced integration may also introduce numerical instabilities, which then require an additional stabilization term.

By default, the **Reduced integration** checkbox is cleared, except in interfaces designed for explicit dynamic analysis, where it is always selected.

It is possible to define different hourglass stabilization methods per mesh element type, for instance, when combining hexahedral and prisms elements within the same domain.

Select a method for **Hourglass stabilization** — **Automatic**, **Energy sampling**, **Hessian**, **Flanagan–Belytschko**, **Manual**, or **None** to be used in combination with the reduced integration scheme.

The **Energy sampling** and **Hessian** methods are available in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Flanagan–Belytschko** method is only available in the Solid Mechanics, Explicit Dynamics interface and should only be used with the Explicit Dynamics study step.

The **Automatic** option selects the stabilization method and its properties based on the physics interface, space dimension, shape function type and order of the displacement field, and the study type.

For the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods, the hourglass stiffness can be scaled by specifying a stiffness multiplier f_{stb}. This multiplier can be an expression of any parameter or variable in the model. This multiplier can be expressed as any parameter or variable in the model. However, to maintain computational efficiency, avoid expensive definitions such as nonlocal coupling operators.

For the hourglass stabilization methods, the stiffness is automatically adjusted during the solution process using an internal multiplier. This accounts for the effect of inelastic deformations, such as damage and plasticity. This correction can be disabled by setting **Inelastic deformations** to **Ignore**. Ignoring inelastic deformations may be necessary for scenarios like cyclic loading, where the hourglass stiffness could become too small during unloading. Alternatively, adjust the value for the **Minimum stiffness multiplier**., which acts as a lower bound for the hourglass stiffness, which is applied in addition to any expression entered for f_{stb}.

For the **Energy sampling** method, the default stabilization uses an isotropic linear elastic potential. Set the **Energy Sampling Potential** to **Hyperelastic** for large deformations. When selected, a neo-Hookean potential is used for nonlinear strains, while linear strains use the linear elastic potential.

The **Hessian** method evaluates a weak contribution using the reduced integration order. For hexahedral mesh elements, this approach may not suppress all hourglass modes. In most cases, non-stabilized modes are suppressed by constraints adjacent to the domain, but if they are not, select **Use full integration** to increase the integration order of the

stabilization equations to suppress all modes. The higher integration order will increase the computational cost of the stabilization method.

- When using the **Manual** option, select **Shear stabilization** or **Volumetric stabilization**. When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.
- When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

ADVANCED

Select how to define **Equivalent elastic constants** — **Automatic**, or **User defined**. When **Automatic** is selected, the equivalent elastic constants are computed from the output of the external material. In some cases this may lead to unexpected results or expensive evaluations when used in for example contact conditions. It is possible to instead manually define the equivalent elastic constants by selecting the **User defined** option. When selected enter values for the equivalent bulk modulus K_{eq} and the equivalent shear modulus G_{eq}.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Material Models** > **External Stress–Strain Relation**
Membrane > **Material Models** > **External Stress–Strain Relation**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **External Stress–Strain Relation**

Physics tab with **Membrane** selected:

Boundaries > **Material Models** > **External Stress–Strain Relation**

Rigid Material

Add the **Rigid Material** node and select one or more geometrical objects to make them a rigid body. **Rigid Material** is a material model, with only one material property: the mass density. It can be used for

- Domains in the Solid Mechanics interface (3D, 2D, and 2D axisymmetry).
- Boundaries in the Shell interface.
- Boundaries (2D) and edges (3D) in the Beam interface.

By default, an **Initial Values** node is added (see Initial Values (Rigid Material)).

You can add functionality to a rigid domain through the following subnodes:

- Fixed Constraint (Rigid Material) to fully constrain the rigid domain.
- Prescribed Displacement/Rotation to prescribe the displacement of individual degrees of freedom.
- Applied Force (Rigid Material) to apply a force in given point.
- Applied Moment (Rigid Material) to apply a moment.
- Mass and Moment of Inertia (Rigid Material) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Material) to add a translational or rotational spring or damper in a given point.

The **Rigid Material** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

In 2D axisymmetric interfaces, the rigid domain has only a single degree of freedom: translation along the Z-axis. In this case it is not possible to give moment loads, mass moment of inertia, or rotational springs.

SHELL PROPERTIES

This section is only present when **Rigid Material** is used in the Layered Shell interface. See the documentation for the Rigid Material node in the Layered Shell chapter.

DENSITY

The default **Density** ρ is taken **From material**. In this case the material assignment for the domain supplies the mass density. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or an expression for the reference temperature locally.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

CENTER OF ROTATION

Select a **Center of Rotation** — **Center of mass**, **Centroid of selected entities**, or **User defined**. The center of rotation affects how displacements are interpreted, and is also used as the default in various subnodes.

- For **Center of mass**, the center of rotation is taken as the center of mass of the rigid domain.
- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The center of rotation is located at the centroid of the selected entities, which do not

need to be related to rigid domain itself. As a special case, you can select a single point, and thus use that point as center of rotation.

Once chosen, a default Center of Rotation: Boundary, Center of Rotation: Edge, or Center of Rotation: Point subnode is automatically added.

- For **User defined**, enter the **Global coordinates of center of rotation**, $\mathbf{X}_c$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{P}} = \mathbf{X}_{\text{P,input}} + \mathbf{X}_{\text{offset}}$$

This section is not available in 2D axisymmetric components.

FORMULATION

Some contributions from a rigid domain will, under geometric nonlinearity, result in a nonsymmetric local stiffness matrix. If all other aspects of the model are such that the global stiffness matrix would be symmetric, then such a nonsymmetric contribution may have a heavy impact on the total solution time and memory usage. In such cases, it is often more efficient to use an approximative local stiffness matrix that is symmetric.

Select **Use symmetric formulation for geometric nonlinearity** to force all matrix contributions from the rigid domain and its subnodes to be symmetric.

Using an approximative stiffness matrix will in general require more iterations. However, since the computational cost per iteration will be cut at least by a factor of two if a symmetric matrix can be used, it is usually more efficient to ignore a weak nonsymmetry.

In particular, if the rotation of the rigid domain per time step or parameter increment is small, there will in general be no increase in the number of iterations at all if this option is used.

When the global stiffness matrix is nonsymmetric for other reasons, then there is nothing to be gained from symmetrizing the contribution from the rigid domain.

CONSTRAINT SETTINGS

When a rigid domain shares a boundary with a flexible material, all nodes on that boundary are constrained to move as a rigid body. As a default these constraints are implemented as pointwise constraints. If you want to use a weak constraint formulation, select **Use weak constraints for rigid–flexible connection**.

ADVANCED

Select **Group dependent variables in solver** — **From physics interface** (default), **Yes**, or **No**, to choose how to group the dependent variables added by the **Rigid Material** feature in the solver sequence.

Rigid Material

Modeling Rigid Bodies: Application Library path
Structural_Mechanics_Module/Connectors_and_Mechanisms/rigid_domain

LOCATION IN USER INTERFACE

Context Menus
Solid Mechanics > **Material Models** > **Rigid Material**
Shell > **Material Models** > **Rigid Material**

Layered Shell > **Material Models** > **Rigid Material**
Beam > **Material Models** > **Rigid Material**

Ribbon
Physics tab with **Solid Mechanics** selected:

Domains > **Material Models** > **Rigid Material**

Physics tab with **Shell** or **Layered Shell** selected:

Boundaries > **Material Models** > **Rigid Material**

Physics tab with **Beam** selected:

Edges > **Material Models** > **Rigid Material** (3D)
Boundaries > **Material Models** > **Rigid Material** (2D)

Initial Values (Rigid Material)

The **Initial Values** node is automatically added as a default node to a Rigid Material node. You can enter initial values for the rigid body displacement, rotation, and velocities, which can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. The initial values that you specify are interpreted in the selected coordinate system.

CENTER OF ROTATION

Select an option from the list: **From parent**, **Centroid of selected entities**, or **User defined**. The given initial values are interpreted at the center of rotation.

- For **From parent**, the center of rotation is taken as the one defined in the parent **Rigid Material** node.
- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The initial conditions act at the centroid of the selected entities, which do not need to be related to rigid domain itself. As a special case, you can select a single point, and thus prescribe the initial conditions at that point

Once chosen, a default Center of Rotation: Boundary, Center of Rotation: Edge, or Center of Rotation: Point subnode is automatically added.

- For **User defined**, enter the **Global coordinates of center of rotation**, $\mathbf{X}_c$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{P}} = \mathbf{X}_{\text{P,input}} + \mathbf{X}_{\text{offset}}$$

INITIAL VALUES: TRANSLATIONAL

- **Displacement at center of rotation** $\mathbf{u}$.
- **Velocity at center of rotation** $\partial\mathbf{u}/\partial t$.

INITIAL VALUES: ROTATIONAL

- *For 3D components only*: **Axis of rotation** Ω
- **Angle of rotation** ϕ.
- **Angular velocity** ω (3D components) and $\partial\phi/\partial t$ (2D components)

LOCATION IN USER INTERFACE

This node is automatically added when the **Rigid Material** node is created. It cannot be added or removed manually.

Fixed Constraint (Rigid Material)

The **Fixed Constraint** node adds a condition that makes the entire rigid domain fixed (fully constrained). The displacements and rotations are zero in all directions. There are no settings for this node. See Rigid Material.

The **Fixed Constraint** and **Prescribed Displacement/Rotation** subnodes cannot be combined for a **Rigid Material**.

REACTION FORCE SETTINGS

Select **Evaluate reaction forces using weak constraints** to compute the reaction force caused by the constraint. The default is to not compute the reaction force. When selected, the constraint is implemented as a weak constraint.

Select **Apply reaction only on rigid body variables** to use a unidirectional constraint for enforcing the prescribed motion. The default is that bidirectional constraints are used. For a fixed constraint, there is no effect of changing this setting.

You can activate and deactivate the fixed constraint by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Fixed Constraint**
Shell > **Rigid Material** > **Fixed Constraint**
Layered Shell > **Rigid Material** > **Fixed Constraint**
Beam > **Rigid Material** > **Fixed Constraint**
Pipe Mechanics > **Rigid Material** > **Fixed Constraint**
Multibody Dynamics > **Rigid Material** > **Fixed Constraint**

Ribbon

Physics tab with **Rigid Material** node selected in the model tree:

Attributes > **Fixed Constraint**

When this node has been added once, it is no longer available for selection in the context menu or ribbon.

Prescribed Displacement/Rotation

The **Prescribed Displacement/Rotation** subnode can be used to selectively constrain or prescribe the degrees of freedom of a **Rigid Material**:

- One or several displacement components can be constrained or prescribed at an arbitrary point.
- One or several rotation components can be constrained.
- The rotation around a given axis can be prescribed.

The prescribed values that you specify are interpreted in the selected coordinate system.

The **Fixed Constraint** and **Prescribed Displacement/Rotation** subnodes cannot be combined for a **Rigid Material**.

CENTER OF ROTATION

The selection of the center of rotation only affects the prescribed displacements, not the rotations.

Select an option from the list: **From parent**, **Centroid of selected entities**, or **User defined**.

- For **From parent**, the center of rotation is taken as the one defined in the parent **Rigid Material** node.
- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The constraints act at the centroid of the selected entities, which do not need to be related to rigid domain itself. As a special case, you can select a single point, and thus constrain that point

Once chosen, a default Center of Rotation: Boundary, Center of Rotation: Edge, or Center of Rotation: Point subnode is automatically added.

- For **User defined**, enter the **Global coordinates of center of rotation**, $\mathbf{X}_c$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{P}} = \mathbf{X}_{\text{P,input}} + \mathbf{X}_{\text{offset}}$$

PRESCRIBED DISPLACEMENT AT CENTER OF ROTATION

Select one or several of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** (3D components only) checkboxes. Then enter a value or an expression for u_{0x}, u_{0y}, and u_{0z} (3D components).

PRESCRIBED ROTATION

Select an option from the **By** list: **Free**, **Constrained rotation**, or **Prescribed rotation**.

For 2D components, and if **Prescribed rotation** is selected, select **Prescribed motion through** — **Angle of rotation** or **Angular velocity**. For **Angle of rotation**, enter the angle ϕ_0. For **Angular velocity**, enter the angular velocity ω_0.

For 3D components:

- For **Constrained rotation** select one or several of the **Constrain rotation around x-axis**, **Constrain rotation around y-axis**, and **Constrain rotation around z-axis** checkboxes.
- For **Prescribed rotation**, enter an **Axis of rotation** vector, Ω. Then, select **Prescribed motion through** — **Angle of rotation** or **Angular velocity**. For **Angle of rotation**, enter the angle ϕ_0. For **Angular velocity**, enter the angular velocity ω_0.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group.
- You can assign the value of the prescribed displacement and rotation to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

REACTION FORCE SETTINGS

Select **Evaluate reaction forces using weak constraints** to compute the reaction force caused by the prescribed motion. The default is to not compute the reaction force. When selected, the prescribed motion is implemented as a weak constraint.

Select **Apply reaction only on rigid body variables** to use a unidirectional constraint for enforcing the prescribed motion. The default is that bidirectional constraints are used. This setting is useful in a situation where a bidirectional constraint would give an unwanted coupling in the equations. This would happen if the prescribed value of the motion is a variable solved for in other equations.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Prescribed Displacement/Rotation**
Shell > **Rigid Material** > **Prescribed Displacement/Rotation**
Layered Shell > **Rigid Material** > **Prescribed Displacement/Rotation**
Beam > **Rigid Material** > **Prescribed Displacement/Rotation**
Pipe Mechanics > **Rigid Material** > **Prescribed Displacement/Rotation**
Multibody Dynamics > **Rigid Material** > **Prescribed Displacement/Rotation**

Ribbon

Physics tab with **Rigid Material** node selected in the model tree:

Attributes > **Prescribed Displacement/Rotation**

When this node has been added once, it is no longer available for selection in the context menu or ribbon.

Applied Force (Rigid Material)

The **Applied Force** subnode is used to apply forces on a rigid domain. The force can act at an arbitrary position in space. A force implicitly contributes also to the moment if it is not applied at the center of mass of a rigid domain. The force that you specify is interpreted in the selected coordinate system.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model. Select a **Coordinate system** for specifying the directions of the force.

LOCATION

Select an option from the list: **Center of rotation**, **Centroid of selected entities**, or **User defined**. This is the location where the force is applied.

- For **Center of rotation**, the location of the load is taken as the center of rotation as defined in the parent **Rigid Material** node.
- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The force acts at the centroid of the selected entities, which do not need to be related to rigid

domain itself. As a special case, you can select a single point, and thus place the force at that point.

Once chosen, a default Location: Boundary, Location: Edge, or Location: Point subnode is automatically added.

- For **User defined**, enter the **Location (global coordinates)**, $\mathbf{X}_p$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_{offset}.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$

APPLIED FORCE

Select the **Direction** of the applied force — **Space-fixed direction** or **Body-fixed direction**.

- For **Space-fixed direction**, the directions of the applied force $\mathbf{F}$ are fixed with respect to the selected **Coordinate system**.
- For **Body-fixed direction**, the directions of the applied force $\mathbf{F}$ follow the rotation of the rigid domain.

Enter values or expressions for the components of the Applied force $\mathbf{F}$. The direction coordinate names can vary depending on the selected coordinate system.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Applied Force**
Shell > **Rigid Material** > **Applied Force**
Layered Shell > **Rigid Material** > **Applied Force**
Beam > **Rigid Material** > **Applied Force**
Pipe Mechanics > **Rigid Material** > **Applied Force**
Multibody Dynamics > **Rigid Material** > **Applied Force**

Ribbon

Physics tab with **Rigid Material** node selected in the model tree:

Attributes > **Applied Force**

Location Nodes

LOCATION: BOUNDARY

The **Location: Boundary** subnode is used to select a set of boundaries whose centroid represents the point of application of a force on a rigid domain.

LOCATION: EDGE

The **Location: Edge** subnode is used to select a set of edges whose centroid represents the point of application of a force on a rigid domain.

LOCATION: POINT

The **Location: Point** subnode is used to select a set of points whose centroid represents the point of application of a force on a rigid domain.

LOCATION IN USER INTERFACE

These nodes are automatically added when **Centroid of selected entities** is selected in the parent **Applied Force** node. They cannot be added or removed manually.

Applied Moment (Rigid Material)

Use the **Applied Moment** subnode to apply moments on a rigid domain. The moment that you specify is interpreted in the selected coordinate system.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model. Select a **Coordinate system** for specifying the directions of the moment.

In 2D, the moment is a scalar input and is not affected by the coordinate system selection.

APPLIED MOMENT

For 3D components, select the **Direction** of the applied moment — **Space-fixed direction** or **Body-fixed direction**.

- For **Space-fixed direction**, the directions of the applied moment $\mathbf{M}$ are fixed with respect to the selected **Coordinate system**.
- For **Body-fixed direction**, the directions of the applied moment $\mathbf{M}$ follow the rotation of the rigid domain.

Enter values or expressions for the **Applied moment** $\mathbf{M}$.

- For 3D components, enter the **x**, **y**, and **z** components of $\mathbf{M}$.
- For 2D components, enter the applied moment around the z-axis, Mz.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Applied Moment**
Shell > **Rigid Material** > **Applied Moment**
Layered Shell > **Rigid Material** > **Applied Moment**
Beam > **Rigid Material** > **Applied Moment**
Pipe Mechanics > **Rigid Material** > **Applied Moment**
Multibody Dynamics > **Rigid Material** > **Applied Moment**

Ribbon

Physics tab with **Rigid Material** node selected in the model tree:

Attributes > **Applied Moment**

Mass and Moment of Inertia (Rigid Material)

Use the **Mass and Moment of Inertia** subnode to add the effect of an associated abstract rigid domain, which is physically not modeled and whose inertial properties are known. These inertial properties can be specified at an arbitrary point in space, which is assumed to be the center of gravity of this extra mass. The moment of inertia tensor that you specify is interpreted in the selected coordinate system.

CENTER OF MASS

Here you specify the location of the center of mass for the contribution given in this node. Select an option from the list — **Center of rotation**, **Centroid of selected entities**, or **User defined**.

- For **Center of rotation**, the location of the load is taken as the center of rotation as defined in the parent **Rigid Material** node.

- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The center of mass is located at the centroid of the selected entities, which do not need to be related to rigid domain itself. As a special case, you can select a single point, and thus position the mass at that point

Once chosen, a default Center of Mass: Boundary, Center of Mass: Edge, or Center of Mass: Point subnode is automatically added.

- For **User defined**, enter the **Global coordinates of center of mass**, $\mathbf{X}_\mathrm{m}$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_offset.

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_\mathrm{m} = \mathbf{X}_\mathrm{m,input} + \mathbf{X}_\mathrm{offset}$$

MASS AND MOMENT OF INERTIA

Enter values or expressions for the **Mass** m. Then for the **Moment of inertia** the axis directions of the moment of inertia tensor are given by the selected coordinate system.

- For 3D components, select **Isotropic**, **Diagonal**, or **Symmetric** and enter one or more components for the tensor I.
- For 2D components, enter a value or an expression for I_z.

FRAME ACCELERATION FORCES

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Mass and Moment of Inertia**
Shell > **Rigid Material** > **Mass and Moment of Inertia**
Layered Shell > **Rigid Material** > **Mass and Moment of Inertia**
Beam > **Rigid Material** > **Mass and Moment of Inertia**
Pipe Mechanics > **Rigid Material** > **Mass and Moment of Inertia**
Multibody Dynamics > **Rigid Material** > **Mass and Moment of Inertia**
Multibody Dynamics > **Spur Gear** > **Mass and Moment of Inertia**
Multibody Dynamics > **Helical Gear** > **Mass and Moment of Inertia**
Multibody Dynamics > **Bevel Gear** > **Mass and Moment of Inertia**
Multibody Dynamics > **Worm Gear** > **Mass and Moment of Inertia**
Multibody Dynamics > **Spur Rack** > **Mass and Moment of Inertia**
Multibody Dynamics > **Helical Rack** > **Mass and Moment of Inertia**

Ribbon

Physics tab with **Rigid Material**, or **Gear** node selected in the model tree:

Attributes > **Mass and Moment of Inertia**

Center of Mass Nodes

CENTER OF MASS: BOUNDARY

Use the **Center of Mass: Boundary** subnode to select a set of boundaries whose centroid represents the center of mass of a Mass and Moment of Inertia (Rigid Material).

CENTER OF MASS: EDGE

The **Center of Mass: Edge** subnode to select a set of edges whose centroid represents the center of mass of a Mass and Moment of Inertia (Rigid Material).

CENTER OF MASS: POINT

The **Center of Mass: Point** subnode to select a set of points whose centroid represents the center of mass of a Mass and Moment of Inertia (Rigid Material).

LOCATION IN USER INTERFACE

These nodes are automatically added when **Centroid of selected entities** is selected in the parent Mass and Moment of Inertia (Rigid Material) node. They cannot be added or removed manually.

Spring Foundation (Rigid Material)

The **Spring Foundation** subnode is used to specify a spring or damper connecting the rigid domain to a fixed ground. The spring can act at an arbitrary position in space. A translational spring implicitly contributes also to the moment if it is not applied at the center of mass of a rigid domain. The data that you specify is interpreted in the selected coordinate system.

LOCATION

Select an option from the list: **Center of rotation**, **Centroid of selected entities**, or **User defined**. This is the location where the spring will be attached.

- For **Center of rotation**, the location of the spring is taken as the center of rotation as defined in the parent **Rigid Material** node.
- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The spring is attached at the centroid of the selected entities, which do not need to be related to the rigid domain itself. As a special case, you can select a single point, and thus attach the spring at that point.

Once chosen, a default Location: Boundary, Location: Edge, or Location: Point subnode is automatically added.

- For **User defined**, enter the **Global coordinates of center of rotation**, $\mathbf{X}_c$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_{offset}.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{P}} = \mathbf{X}_{\text{P,input}} + \mathbf{X}_{\text{offset}}$$

SPRING

Select a **Spring type** — **Spring constant** or **Force as function of extension**.

When **Spring constant** is selected, the translational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Force as function of extension** is selected, enter the force vector $\mathbf{F}_s$. It must be a function of the built-in variables describing the spring extension. The default value indicates the correct variable name, for example, `solid.rd1.uspring1_spf1`.

ROTATIONAL SPRING

Select a **Spring type** — **Spring constant** or **Moment as function of rotation**.

When **Spring constant** is selected, the rotational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Moment as function of rotation** is selected, enter the moment vector $\mathbf{M}_s$. It must be a function of the built-in variables describing the spring extension. The default value indicates the correct variable name, for example, `solid.rd1.thspring1_spf1`.

In 2D the **Spring constant** is only a single scalar, representing the stiffness for rotation around the out-of-plane direction. Similarly, the **Moment as function of rotation** is a single scalar.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** $\eta_{u,s}$, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions in the table for the **Loss factor for spring** $\eta_{u,k}$ or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

All settings in this section are analogous to the corresponding settings in the Loss Factor Damping section. In 2D, only one scalar loss factor, corresponding to rotation around the out-of-plane axis, is given.

VISCOUS DAMPING

Select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions for the damping constants d_u in the table. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS ROTATIONAL DAMPING

All settings in this section are analogous to the corresponding settings in the Viscous Damping section. In 2D, only one scalar viscous damping, corresponding to rotation around the out-of-plane axis, is given.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Material** > **Spring Foundation**
Shell > **Rigid Material** > **Spring Foundation**
Layered Shell > **Rigid Material** > **Spring Foundation**
Beam > **Rigid Material** > **Spring Foundation**
Pipe Mechanics > **Rigid Material** > **Spring Foundation**
Multibody Dynamics > **Rigid Material** > **Spring Foundation**

Ribbon

Physics tab with **Rigid Material** node selected in the model tree:

Attributes > **Spring Foundation**

Free

The **Free** node is the default boundary condition. It means that there are no constraints and no loads acting on the boundary. When the physics interface is added, a default **Free** node is added. If you look at the selections for this node, it will show all boundaries that do not have any boundary conditions applied.

In each physics interface, the **Free** node applies to a geometric entity that is one level below the one on which the physics is active:

- For the Solid Mechanics, Plate, and Multibody Dynamics interfaces, the **Free** node applies to boundaries.
- For the Shell, Layered Shell, and Membrane interfaces, the **Free** node applies to edges.
- For the Beam, Pipe Mechanics, and Truss interfaces, the **Free** node applies to points.

You can manually add **Free** nodes to override other boundary conditions. This is however seldom needed.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More Constraints** > **Free**
Solid Mechanics, Explicit Dynamics > **More Constraints** > **Free**
Shell > **More Constraints** > **Free**
Plate > **More Constraints** > **Free**
Layered Shell > **More Constraints** > **Free**
Membrane > **More Constraints** > **Free**
Beam > **More Constraints** > **Free**
Pipe Mechanics > **More Constraints** > **Free**
Truss > **More Constraints** > **Free**
Truss, Explicit Dynamics > **More Constraints** > **Free**
Wire > **More Constraints** > **Free**
Multibody Dynamics > **More Constraints** > **Free**

Ribbon

Physics tab with a physics interface selected:

Boundaries > **More Constraints** > **Free**
Edges > **More Constraints** > **Free**
Points > **More Constraints** > **Free**

Prescribed Displacement

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in one or more directions to the geometric entity (domain, boundary, edge, or point). It is also possible set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

If a displacement is prescribed in one direction, this leaves the structure free to deform in the orthogonal directions.

If a zero displacement is applied in all directions, this is the same as a Fixed Constraint.

For details, see Prescribed Displacements, Velocities, and Accelerations.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Displacement node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Displacement, Interface node.

COORDINATE SYSTEM SELECTION

The constraints are applied in the axis directions of the coordinate system selected in this section. You should only use a coordinate system that has fixed spatial directions.

PRESCRIBED DISPLACEMENT

For the displacement in each direction, select a setting from the list — **Free**, **Prescribed**, or **Limited**. Select:

- **Free** (the default) to leave the displacement component unconstrained.
- **Prescribed** to constrain the displacement component to a given value. Enter a scalar value for the component of the prescribed displacement u_0.
- **Limited** to set a maximum and a minimum limit for the displacement component. Enter values for the maximum displacement $u_{0,\max}$ and the minimum displacement $u_{0,\min}$. By default, they are set to `Inf` and `-Inf`, which corresponds to no active constraint.

If any displacement component is set to **Limited**, an additional section **Limited Displacement** is visible. Select the **Method** used to implement the weak inequality

constraint — **Penalty** or **Augmented Lagrangian**. For both methods, enter a **Penalty factor** k_p.

The option **Augmented Lagrangian** is not available in physics interfaces for time-explicit dynamics.

By default, the **Penalty** method is suggested, which in principle enforces the maximum and minimum limits for the displacement by adding nonlinear springs with a stiffness equal to k_p when the limits are exceeded. This method is usually robust, but the accuracy is directly dependent on the chosen penalty factor.

The **Augmented Lagrangian** method adds extra degrees of freedom to improve the accuracy of the constraint. Here, the penalty factor is a numerical parameter, and has less impact on the accuracy of the constraint compared to when using the penalty method. The implementation of the augmented Lagrangian method puts no restrictions on the solver sequence, but for good convergence, proper scaling of the extra degrees of freedom can be important.

The default value for the **Penalty factor** k_p depends on the physics interface, and on what type of entity the **Prescribed Displacement** node is added. In the expressions below, `<phys>` is the tag of the physics interface, and `<pd>` is the tag of the **Prescribed Displacement** node.

Solid Mechanics, Multibody Dynamics, Layered Shell, and Membrane

- For points, the default expression is `100*<phys>.Eequ*<phys>.<pd>.charLen`
- For edges, the default expression is `100*<phys>.Eequ`
- For boundaries, the default expression is `100*<phys>.Eequ/<phys>.<pd>.charLen`
- For domains, the default expression is `100*<phys>.Eequ/<phys>.<pd>.charLen^2`

Above, the terms point, edge, boundary, and domain refer to the actual *physical* entities represented by the chosen *geometric* entities. For instance, in a 2D solid the out-of-plane thickness is only implicitly modeled, and thus the physical entity level is interpreted as one level higher than the chosen geometric entity level.

The variable `<phys>.<pd>.charLen` is by default equal to the length of the mesh element for the Solid Mechanics and Multibody Dynamics interfaces. For the Layered Shell and Membrane interfaces, `<phys>.<pd>.charLen` is the thickness.

Truss

- For points, the default expression is `100*truss.Eequ*truss.area/truss.<pd>.charLen`
- For edges, the default expression is `100*truss.Eequ*truss.area/truss.<pd>.charLen^2`

The variable `truss.<pd>.charLen` is by default equal to the length of the mesh element.

Wire

- For points, the default expression is `100*wire.k_A/wire.<pd>.charLen`
- For edges, the default expression is `100*wire.k_A/wire.<pd>.charLen^2`

The variable `wire.<pd>.charLen` is by default equal to the length of the mesh element.

For details about setting maximum and minimum limits for the displacements, see Limited Displacement

In versions prior to 6.2, you could chose to formulate the constraint in two ways: standard notation or general notation. Since it was rarely used, the general notation option is no longer supported. There are more natural ways of specifying identical constraints.

If you have a model created in version 6.1 or earlier, where general notation was used, the settings for this option are visible, and are still functional. You should, however, consider to reformulate the constraints, since the general notation may no longer be supported in future versions.

For reference, the documentation for the general notation is provided below:

You can specify the displacements using a general notation that includes any linear combination of displacement components. For example, for 2D components, use the relationship

$$H\begin{bmatrix} u \\ v \end{bmatrix} = R$$

For **H** matrix H select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and then enter values as needed in the field or matrix. Enter values or expressions for the **R** vector R.

For example, to achieve the condition $u = v$, use the settings

$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which force the domain to move only diagonally in the xy-plane.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

When **Individual dependent variables** is selected in the **Apply reaction terms on** list, the constraint forces are applied directly on the degrees of freedom, which are the displacements along the global coordinate axes. If you use this setting together with a local coordinate system, the results will be inconsistent since the constraint forces will not match the constraint orientation.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group.
- You can assign the value of the prescribed displacement to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Prescribed Displacement**
Solid Mechanics > **Prescribed Displacement** (Boundary)
Solid Mechanics > **Edges** > **Prescribed Displacement**
Solid Mechanics > **Points** > **Prescribed Displacement**
Solid Mechanics > **Thin Layer** > **Prescribed Displacement**
Solid Mechanics, Explicit Dynamics > **Prescribed Displacement** (Boundary)
Solid Mechanics, Explicit Dynamics > **Edges** > **Prescribed Displacement**

Solid Mechanics, Explicit Dynamics > **Points** > **Prescribed Displacement**
Layered Shell > **Face and Domain Constraints** > **Prescribed Displacement**
Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Displacement, Interface**
Layered Shell > **Prescribed Displacement** (Edge)
Layered Shell > **More Constraints** > **Interfaces** > **Prescribed Displacement, Interface** (Edge)
Layered Shell > **Points** > **Prescribed Displacement**
Layered Shell > **Points** > **Interfaces** > **Prescribed Displacement, Interface**
Membrane > **Face Constraints** > **Prescribed Displacement**
Membrane > **Prescribed Displacement** (Edge)
Membrane > **Points** > **Prescribed Displacement**
Truss > **Line Constraints** > **Prescribed Displacement**
Truss > **Prescribed Displacement** (Point)
Truss, Explicit Dynamics > **Prescribed Displacement** (Point)
Wire > **Line Constraints** > **Prescribed Displacement**
Wire > **Prescribed Displacement** (Point)
Multibody Dynamics > **Domain Constraints** > **Prescribed Displacement**
Multibody Dynamics > **More Constraints** > **Prescribed Displacement**
Multibody Dynamics > **Edges** > **Prescribed Displacement**
Multibody Dynamics > **Points** > **Prescribed Displacement**

Ribbon
Physics tab with **Solid Mechanics** selected:

Domains > **Domain Constraints** > **Prescribed Displacement**
Boundaries > **Solid Mechanics** > **Prescribed Displacement**
Edges > **Solid Mechanics** > **Prescribed Displacement**
Points > **Solid Mechanics** > **Prescribed Displacement**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **Solid Mechanics, Explicit Dynamics** > **Prescribed Displacement**
Edges > **Solid Mechanics, Explicit Dynamics** > **Prescribed Displacement**
Points > **Solid Mechanics, Explicit Dynamics** > **Prescribed Displacement**

Physics tab with **Membrane** selected:

Boundaries > **Face Constraints** > **Prescribed Displacement**
Edges > **Membrane** > **Prescribed Displacement**
Points > **Membrane** > **Prescribed Displacement**

Physics tab with **Truss** selected:

Edges > **Line Constraints** > **Prescribed Displacement**
Points > **Truss** > **Prescribed Displacement**

Physics tab with **Truss, Explicit Dynamics** selected:

Points > **Truss, Explicit Dynamics** > **Prescribed Displacement**

Physics tab with **Wire** selected:

Edges > **Line Constraints** > **Prescribed Displacement**
Points > **Wire** > **Prescribed Displacement**

Physics tab with **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Prescribed Displacement**
Boundaries > **More Constraints** > **Prescribed Displacement**
Edges > **Multibody Dynamics** > **Prescribed Displacement**
Points > **Multibody Dynamics** > **Prescribed Displacement**

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Prescribed Displacement** (Boundary)

Prescribed Velocity

The **Prescribed Velocity** node adds a boundary or domain condition where the velocity is prescribed in one or more directions. The prescribed velocity condition is applicable for Time Dependent and Frequency Domain studies. It is possible to prescribe a velocity in one direction, leaving the solid free in the other directions. The **Prescribed Velocity** node is a constraint and overrides any other constraint on the same selection. The **Prescribed Velocity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Velocity and node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Velocity, Interface node.

COORDINATE SYSTEM SELECTION

Select a coordinate system in which the given velocity components will be interpreted.

Coordinate systems with directions that change with time should not be used for a prescribed velocity.

PRESCRIBED VELOCITY

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** checkboxes (only applicable checkboxes are shown based on the spatial dimension). Then enter a value or an expression for v_x, v_y, and v_z. For axisymmetric components, select one or both of the **Prescribed in *r* direction** and **Prescribed in z direction** checkboxes. Then enter a value or an expression for v_r and v_z. When twist is included in 2D axisymmetry, the circumferential displacement can also be prescribed.

In a stationary study, this node can either be ignored or treated as a constraint (similar to a **Prescribed Displacement** node with zero displacement). To control this, select an option from the **Displacement in stationary study** list — **Free** or **Constrained**.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Prescribed Velocity**

Solid Mechanics > **More Constraints** > **Prescribed Velocity** (Boundary)

Layered Shell > **Face and Domain Constraints** > **Prescribed Velocity**

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Velocity, Interface**
Layered Shell > **More Constraints** > **Prescribed Velocity** (Edge)
Membrane > **Face Constraints** > **Prescribed Velocity**
Membrane > **More Constraints** > **Prescribed Velocity** (Edge)
Truss > **Line Constraints** > **Prescribed Velocity**
Truss > **More Constraints** > **Prescribed Velocity** (Point)
Multibody Dynamics > **Domain Constraints** > **Prescribed Velocity**
Multibody Dynamics > **More Constraints** > **Prescribed Velocity**
Multibody Dynamics > **Edges** > **Prescribed Velocity**
Multibody Dynamics > **Points** > **Prescribed Velocity**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Domain Constraints** > **Prescribed Velocity**
Boundaries > **More Constraints** > **Prescribed Velocity**

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Prescribed Velocity**
Boundaries > **Interfaces** > **Prescribed Velocity, Interface**
Edges > **More Constraints** > **Prescribed Velocity**

Physics tab with **Membrane** selected:

Boundaries > **Face Constraints** > **Prescribed Velocity**
Edges > **More Constraints** > **Prescribed Velocity**

Physics tab with **Truss** or **Wire** selected:

Edges > **Line Constraints** > **Prescribed Velocity**
Points > **More Constraints** > **Prescribed Velocity**

Physics tab with **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Prescribed Velocity**
Boundaries > **More Constraints** > **Prescribed Velocity**
Edges > **Multibody Dynamics** > **Prescribed Velocity**
Points > **Multibody Dynamics** > **Prescribed Velocity**

Prescribed Acceleration

The **Prescribed Acceleration** node adds a boundary or domain condition, where the acceleration is prescribed in one or more directions. The prescribed acceleration

condition is applicable for Time Dependent and Frequency Domain studies. It is possible to prescribe an acceleration in one direction, leaving the solid free in the other directions. The **Prescribed Acceleration** node is a constraint and overrides any other constraint on the same selection.

The **Prescribed Acceleration** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Acceleration and node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Prescribed Acceleration, Interface node.

COORDINATE SYSTEM SELECTION

Select a coordinate system in which the given acceleration components will be interpreted.

Coordinate systems with directions that change with time should not be used for a prescribed acceleration.

PRESCRIBED ACCELERATION

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** checkboxes (only applicable checkboxes are shown based on the spatial dimension). Then enter a value or an expression for a_x, a_y, and a_z. For axisymmetric components, select one or both of the **Prescribed in r direction** and **Prescribed in z**

direction checkboxes. Then enter a value or an expression for a_r and a_z. When twist is included in 2D axisymmetry, the circumferential displacement can also be prescribed.

In a stationary study, this node can either be ignored or treated as a constraint (similar to a **Prescribed Displacement** node with zero displacement). To control this, select an option from the **Displacement in stationary study** list — **Free** or **Constrained**.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Prescribed Acceleration**
Solid Mechanics > **More Constraints** > **Prescribed Acceleration** (Boundary)
Layered Shell > **Face and Domain Constraints** > **Prescribed Acceleration**
Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Acceleration, Interface**
Layered Shell>More Constraints>Prescribed Acceleration (Edge)
Membrane > **Face Constraints** > **Prescribed Acceleration**
Membrane > **More Constraints** > **Prescribed Acceleration** (Edge)
Truss > **Line Constraints** > **Prescribed Acceleration**
Truss > **More Constraints** > **Prescribed Acceleration** (Point)
Wire > **Line Constraints** > **Prescribed Acceleration**
Wire > **More Constraints** > **Prescribed Acceleration** (Point)
Multibody Dynamics > **Domain Constraints** > **Prescribed Acceleration**
Multibody Dynamics > **More Constraints** > **Prescribed Acceleration**
Multibody Dynamics > **Edges** > **Prescribed Acceleration**
Multibody Dynamics > **Points** > **Prescribed Acceleration**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Domain Constraints** > **Prescribed Acceleration**
Boundaries > **More Constraints** > **Prescribed Acceleration**

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Prescribed Acceleration**
Boundaries > **Interfaces** > **Prescribed Acceleration, Interface**
Edges > **More Constraints** > **Prescribed Acceleration**

Physics tab with **Membrane** selected:

Boundaries > **Face Constraints** > **Prescribed Acceleration**
Edges > **More Constraints** > **Prescribed Acceleration**

Physics tab with **Truss** or **Wire** selected:

Edges > **Line Constraints** > **Prescribed Acceleration**
Points > **More Constraints** > **Prescribed Acceleration**

Physics tab with **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Prescribed Acceleration**
Boundaries > **More Constraints** > **Prescribed Acceleration**
Edges > **Multibody Dynamics** > **Prescribed Acceleration**
Points > **Multibody Dynamics** > **Prescribed Acceleration**

Fixed Constraint

The **Fixed Constraint** node adds a condition that makes the geometric entity fixed (fully constrained); that is, the displacements are zero in all directions on the selected geometrical entities. If there are rotational degrees of freedom, they will also be zero.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Fixed Constraint node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Fixed Constraint, Interface node.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Fixed Constraint**
Solid Mechanics > **Fixed Constraint** (Boundary)
Solid Mechanics > **Edges** > **Fixed Constraint**
Solid Mechanics > **Points** > **Fixed Constraint**
Solid Mechanics > **Thin Layer** > **Fixed Constraint** (Boundary)
Solid Mechanics, Explicit Dynamics > **Domain Constraints** > **Fixed Constraint**
Solid Mechanics, Explicit Dynamics > **Fixed Constraint** (Boundary)
Solid Mechanics, Explicit Dynamics > **Edges** > **Fixed Constraint**
Solid Mechanics, Explicit Dynamics > **Points** > **Fixed Constraint**
Layered Shell > **Face and Domain Constraints** > **Fixed Constraint**
Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Fixed Constraint, Interface**
Layered Shell > **Fixed Constraint** (Edge)
Layered Shell > **More Constraints** > **Interfaces** > **Fixed Constraint, Interface** (Edge)
Layered Shell > **Points** > **Fixed Constraint**
Layered Shell > **Points** > **Interfaces** > **Fixed Constraint, Interface**
Membrane > **Face Constraints** > **Fixed Constraint**
Membrane > **Fixed Constraint** (Edge)
Membrane > **Points** > **Fixed Constraint**
Beam > **Line Constraints** > **Fixed Constraint**

Beam > **Fixed Constraint** (Point)
Multibody Dynamics > **Domain Constraints** > **Fixed Constraint**
Multibody Dynamics > **Fixed Constraint** (Boundary)
Multibody Dynamics > **Edges** > **Fixed Constraint**
Multibody Dynamics > **Points** > **Fixed Constraint**

Ribbon
Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Domain Constraints** > **Fixed Constraint**
Boundaries > **Solid Mechanics** > **Fixed Constraint**
Edges > **Solid Mechanics** > **Fixed Constraint**
Points > **Solid Mechanics** > **Fixed Constraint**

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Fixed Constraint**
Boundaries > **Interfaces** > **Fixed Constraint, Interface**
Edges > **Layered Shell** > **Fixed Constraint**
Edges > **Interfaces** > **Fixed Constraint, Interface**
Points > **Layered Shell** > **Fixed Constraint**
Points > **Interfaces** > **Fixed Constraint, Interface**

Physics tab with **Membrane** selected:

Boundaries > **Face Constraints** > **Fixed Constraint**
Edges > **Membrane** > **Fixed Constraint**
Points > **Membrane** > **Fixed Constraint**

Physics tab with **Beam** selected:

Edges > **Line Constraints** > **Fixed Constraint**
Points > **Truss** > **Fixed Constraint**

Physics tab with **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Fixed Constraint**
Boundaries > **Multibody Dynamics** > **Fixed Constraint**
Edges > **Multibody Dynamics** > **Fixed Constraint**
Points > **Multibody Dynamics** > **Fixed Constraint**

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Fixed Constraint** (Boundary)

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Fixed Constraint** or **Prescribed Displacement**) to prescribe a deformation of the constrained boundary caused by changes in the surrounding temperature. This makes it possible to reduce the stresses caused by such boundary conditions.

Select an **Input type** to specify how the thermal strain is entered. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

here, the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\text{t}}(\tau)d\tau\right) - 1$$

where α_{t} is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

The **Thermal Expansion** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Thermal Expansion (for Constraints) node.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from domain** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation. For this reason, it is only possible to inherit from a **Thermal Expansion** node under a material model, not from the **Thermal Expansion** multiphysics coupling.
- The virtual surrounding material has the same thermal expansion as the domain itself.

The **Inherit from domain** checkbox is not available when the constraint in the parent node is applied to domains. In that case, the data is implicitly inherited from the domain.

When **Inherit from domain** is not selected, enter:

- A value or expression for the **Volume reference temperature** T_{ref} that is the temperature at which there are no thermal displacements at the constraints.
- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any temperature variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The default values **From material** are used. This requires that a material has been assigned to the boundaries, edges, or points where the constraint is active. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several

different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the *Structural Mechanics Modeling* chapter.
- Thermal Expansion of Constraints in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Fixed Constraint** > **Thermal Expansion**
Solid Mechanics > **Prescribed Displacement** > **Thermal Expansion**
Layered Shell > **Fixed Constraint** > **Thermal Expansion**
Layered Shell > **Prescribed Displacement** > **Thermal Expansion**
Membrane > **Fixed Constraint** > **Thermal Expansion**
Membrane > **Prescribed Displacement** > **Thermal Expansion**
Multibody Dynamics > **Fixed Constraint** > **Thermal Expansion**
Multibody Dynamics > **Prescribed Displacement** > **Thermal Expansion**

Ribbon

Physics tab with **Fixed Constraint** or **Prescribed Displacement** node selected in the model tree:

Attributes > **Thermal Expansion**

Roller

The **Roller** node adds a roller constraint as the boundary condition; that is, the displacement is zero in the direction perpendicular (normal) to the boundary, but the boundary is free to move in the tangential direction. A **Roller** condition is similar to a Symmetry condition, but the latter has some other options. The **Roller** constraint is not available in 1D and 1D axisymmetry; use a **Fixed Constraint** instead.

COORDINATE SYSTEM SELECTION

By selecting a coordinate system, you can simplify the description of axis orientations when using the **Plane** or **Cylinder** options in the **Roller Constraint** section.

ROLLER CONSTRAINT

The default roller constraint acts along a normal to the selected boundaries, which is computed numerically. For a certain mesh vertex, this direction is kept constant during the analysis, so it cannot represent finite sliding on a curved boundary. You can however, for some important cases, prescribe that the boundary slides on an analytical surface. There are two situations where you may want to do this:

- The geometry or mesh representation of the actual boundary is imperfect. This can lead to unexpected locking of the deformation, since the computed normals do not have the intended directions.
- In a geometrically nonlinear analysis, the use of an analytical surface makes it possible to model finite sliding also along curved boundaries.

Select a **Normal orientation** — **Automatic**, **Plane**, **Cylinder**, or **Sphere**.

The options **Cylinder** and **Sphere** are not available in the Solid Mechanics, Explicit Dynamics interface.

For **Automatic**, the constraint is applied in the direction normal to the boundary, as represented by the mesh. This orientation is not updated during the analysis.

For **Plane**, select **Orientation of normal** — **X-axis**, **Y-axis**, **Z-axis**, or **User defined** to describe the constraint direction. When **User defined** is used, enter the vector $\mathbf{e}_n$, pointing along the normal to the plane along which the boundary is sliding. The orientation is interpreted in the coordinate system given in the **Coordinate System Selection** section.

For **Cylinder**, select **Axis type** — **X-axis**, **Y-axis**, **Z-axis**, or **User defined** to describe the orientation of the cylinder. When **User defined** is used, enter the vector $\mathbf{e}_s$, pointing along the axis of the cylinder. The orientation is interpreted in the coordinate system given in the **Coordinate System Selection** section. Enter the location of a **Point on axis**, $\mathbf{X}_c$, in order fix the location of the cylinder. The constraints will act radially from the cylinder axis. If the analysis is geometrically nonlinear, each mesh vertex is assumed to maintain its distance from the cylinder axis.

For **Sphere**, enter the location of the **Center of sphere**, $\mathbf{X}_c$. The constraints will act radially from this point. If the analysis is geometrically nonlinear, each mesh vertex is assumed to maintain its distance from the center of the sphere.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the theory section of the *Structural Mechanics User's Guide:*

- Rigid Connector

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Constraint Settings
- Weak Constraints

EXCLUDED EDGES, EXCLUDED POINTS

To display these sections (**Excluded Edges** in 3D geometries only), click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about these sections, see Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Roller**
Solid Mechanics > **Thin Layer** > **Roller** (Boundary)
Solid Mechanics, Explicit Dynamics > **Roller**
Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Roller, Interface**
Layered Shell > **Roller**
Multibody Dynamics > **More Constraints** > **Roller**

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Solid Mechanics** > **Roller**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **Solid Mechanics, Explicit Dynamics** > **Roller**

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Roller, Interface**
Edges > **Layered Shell** > **Roller**

Physics tab with **Multibody Dynamics** selected:

Boundaries > **More Constraints** > **Roller**

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Roller** (Boundary)

Symmetry

The **Symmetry** node adds a boundary condition that represents symmetry in the geometry and in the loads. A symmetry condition is free in the plane and fixed in the out-of-plane direction.

When applied to an edge (in the Membrane interface) the symmetry plane is formed by the normal to the boundary and the edge tangent.

When the model is axisymmetric, use the Symmetry Plane node instead.

The **Symmetry** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). The **Symmetry** constraint is not available in 1D and 1D axisymmetry; use a **Fixed constraint** instead.

The symmetry constraint implies constraining the displacement in the normal direction, $\mathbf{u}\cdot\mathbf{n} = \alpha$. For a standard symmetry condition, $\alpha = 0$, but this can be modified by settings in the **Normal Direction Condition** section.

You can select boundaries that represent different symmetry planes in a single **Symmetry** node. There are however some cases when such a selection should be avoided.

- If all selected boundaries, to within a small tolerance, are located in the same plane, then the normal vector, $\mathbf{n}$, is computed using an averaging scheme. This remedies problems that can occur for example when not all nodes on a symmetry plane of an imported mesh lie exactly on the same plane. However, averaging is not possible if several symmetry planes are selected.
- In the **Constraint Settings** section, you have the possibility to select **Nodal** as **Constraint Method**. The nodal method is more efficient than the default **Elemental** method. However, if two element faces that do not represent the same plane are part of the selection, the generated constraint will be incorrect at their intersection.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Symmetry node.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_n. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un` (for example, `comp1.solid.sym1.un`).

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More Constraints** > **Symmetry**
Solid Mechanics, Explicit Dynamics > **More Constraints** > **Symmetry**
Layered Shell > **More Constraints** > **Symmetry**
Membrane > **More Constraints** > **Symmetry**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More Constraints** > **Symmetry**

Physics tab with **Layered Shell** or **Membrane** selected:

Edges > **More Constraints** > **Symmetry**

Symmetry Plane

The **Symmetry Plane** node adds a boundary condition that represents symmetry in the geometry and in the loads. This node is available only in 2D axisymmetry, where the only possible symmetry plane is normal to the Z-axis.

CIRCUMFERENTIAL CONDITION

This section is shown only if **Include circumferential displacement** has been selected in the settings for a **Solid Mechanics** node.

Select either **Symmetry** or **Antisymmetry**.

With the symmetric assumption, the solution for the circumferential displacement is $v(R, 2Z_p\text{-}Z) = v(R, Z)$, where Z_p is the location of the symmetry plane.

With the antisymmetric assumption, an extra constraint $v=0$ is added in the symmetry plane, giving a solution for the circumferential displacement of the type $v(R, 2Z_p\text{-}Z) = -v(R, Z)$. This makes it possible to model a structure that is geometrically symmetric, but subjected to a twisting state of deformation.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_{n}. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un` (for example, `comp1.solid.symp1.un`).

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More Constraints** > **Symmetry Plane**
Solid Mechanics, Explicit Dynamics > **More Constraints** > **Symmetry Plane**
Membrane > **More Constraints** > **Symmetry Plane**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More Constraints** > **Symmetry Plane**

Physics tab with **Membrane** selected:

Edges > **More Constraints** > **Symmetry Plane**

Antisymmetry

The **Antisymmetry** node adds a boundary condition for an antisymmetry boundary. An antisymmetry condition is fixed in the plane and free in the out-of-plane direction.

When applied to an edge (in the Membrane interface) the antisymmetry plane is formed by the normal to the boundary and the edge tangent.

The **Antisymmetry** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available for 3D and 2D.

In a geometrically nonlinear analysis, large rotations must not occur at the antisymmetry plane because this causes artificial straining.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Antisymmetry node.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More Constraints** > **Antisymmetry**
Solid Mechanics, Explicit Dynamics > **More Constraints** > **Antisymmetry**
Layered Shell > **Antisymmetry**
Membrane > **More Constraints** > **Antisymmetry**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More Constraints** > **Antisymmetry**

Physics tab with **Membrane** or **Layered Shell** selected:

Edges > **More Constraints** > **Antisymmetry**

Rigid Motion Suppression

The **Rigid Motion Suppression** node adds constraints required to suppress rigid body modes. The constraints are selected so that there will be no reaction forces if the external loads are self-equilibrating.

The constraints will, depending on the physics interface and the geometrical dimension, be applied to reference points as needed. These points are automatically picked from the selected geometrical domains.

If you need to constrain several objects that are not physically connected, you must add as many **Rigid Motion Suppression** nodes as there are disconnected objects. **Rigid Motion Suppression** is available for 3D, 2D, and 2D axisymmetry.

The **Rigid Motion Suppression** feature must not be combined with any other type of constraint, such as Prescribed Displacement, Roller, and Symmetry.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Rigid Motion Suppression node.

CONTRIBUTING POINTS

By default the selectable input is **Automatic**, which corresponds to using all points in the domain. When selecting **From selection input**, a new **Point Selection** section is shown where specific points can be selected. For 3D at least three points are expected, and for 2D at least two points are expected.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

- Rigid Body Motion in the *Structural Mechanics Modeling* chapter.
- Rigid Motion Suppression in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Rigid Motion Suppression**
Shell > **Face Constraints** > **Rigid Motion Suppression**
Plate > **Face Constraints** > **Rigid Motion Suppression**
Layered Shell > **Face Constraints** > **Rigid Motion Suppression**
Membrane > **Face Constraints** > **Rigid Motion Suppression**
Beam > **Line Constraints** > **Rigid Motion Suppression**
Truss > **Line Constraints** > **Rigid Motion Suppression**
Multibody Dynamics > **Domain Constraints** > **Rigid Motion Suppression**

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Rigid Motion Suppression**

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Face Constraints** > **Rigid Motion Suppression**

Physics tab with **Plate** selected:

Domains > **Face Constraints** > **Rigid Motion Suppression**

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Rigid Motion Suppression**

Physics tab with **Beam** or **Truss** selected:

Edges > **Line Constraints** > **Fixed Constraint**

Body Load

Add a **Body Load** to domains for modeling general volumetric loads.

For loads caused by gravity or rotation, it is more convenient to use the Gravity and Rotating Frame nodes.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Body Load node.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select a Load type — **Force per reference volume**, **Force per deformed volume**, or **Total force**, then enter values or expressions for the components based on the selection and

the space dimension. For 2D components, it is possible to specify the **Force per reference area** or the **Force per deformed area**.

- After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.
- For **Total force**, COMSOL Multiphysics divides the total force by the volume of the domains where the load is active.
- For 2D components, and if **Force per reference area** or **Force per deformed area** is selected, the body load as force per unit volume is then the value of the force per area divided by the thickness.

TABLE 4-4: BODY LOAD TYPES.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference volume	$\mathbf{f}_V$	N/m^3	domains	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)
Force per deformed volume	$\mathbf{f}_v$	N/m^3	domains	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)
Force per reference area	$\mathbf{f}_A$	N/m^2	domains	2D (x, y)
Force per deformed area	$\mathbf{f}_a$	N/m^2	domains	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	domains	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Volume Forces** > **Body Load**
Solid Mechanics, Explicit Dynamics > **Volume Forces** > **Body Load**
Layered Shell > **Face and Volume Loads** > **Body Load**
Membrane > **Face and Volume Loads** > **Body Load**
Multibody Dynamics > **Volume Forces** > **Body Load**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Domains > **Volume Forces** > **Body Load**

Ribbon

Physics tab with **Layered Shell** or **Membrane** selected:

Boundaries > **Face and Volume Loads** > **Body Load**

Gravity

When you add a **Gravity** node, gravity forces are applied to all features in the physics interface with a density, mass, or mass distribution.

The gravity acts in a fixed spatial direction. The load intensity is $\mathbf{g} = \rho g \mathbf{e}_g$ where g as a default is the acceleration of gravity (one of the predefined Physical Constants).

The **Gravity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

Gravity is not available in 1D axisymmetry.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system must have orthonormal coordinate axes, and be defined in the spatial frame. Select a coordinate system in which the gravity field vector is represented. Only systems with space-independent axis directions are consistent with the assumptions of this feature.

GRAVITY

Enter the components for the acceleration due to **Gravity g**. The default value is `g_const`, which is the gravitational acceleration constant at mean sea level having the value 9.80665 $\mathrm{m/s^2}$.

- For 3D, the default is that the gravity acts in the negative z direction.
- For 2D, the default is that the gravity acts in the negative y direction, except in the Plate interface where the default is in the negative z direction.
- For 1D, the default is that the gravity acts in the negative x direction.
- For 2D axisymmetry, the default is that the gravity acts in the negative z direction.

Frame Acceleration Forces

In versions earlier than 6.1, the **Gravity** node was a feature with domain selection. In a model created in an earlier version, the old **Gravity** node remains unchanged. For such a model, it is recommended to use the newer functionality. There are different options:

- In most cases, switching to the new version of **Gravity** is appropriate.
- If the current **Gravity** node has a selection, which is not the entire model, or if it contains expressions that depends on the spatial coordinates, you can replace it by a Linearly Accelerated Frame node.
- If the current **Gravity** node is used to represent an acceleration imposed by the foundation, consider replacing it by a Base Excitation node.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Gravity**
Solid Mechanics, Explicit Dynamics > **Gravity**
Shell > **Gravity**
Layered Shell > **Gravity**
Plate > **Gravity**
Membrane > **Gravity**
Beam > **Gravity**
Truss > **Gravity**
Truss, Explicit Dynamics > **Gravity**
Wire > **Gravity**
Multibody Dynamics > **Gravity**

Ribbon

Physics tab with a structural mechanics interface selected:

Global > **Gravity**

Base Excitation

The **Base Excitation** node is used to represent an acceleration that is applied at all constraints. The main use case is for modal-based analyses, in which case it is not possible to directly prescribe nonzero displacements at constrained degrees of freedom. It is, however, possible to use **Base Excitation** in any type of dynamic analysis.

When a structure is subjected to a base excitation, displacements, velocities, and accelerations are measured relative to a space-fixed coordinate system.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system must have orthonormal coordinate axes, and be defined in the spatial frame. Select a coordinate system in which the base acceleration vector is represented. Only systems with space-independent axis directions are consistent with the assumptions of the base excitation.

BASE EXCITATION

Enter the **Base acceleration**, $\mathbf{a}_b$.

For postprocessing purposes, the feature also computes the absolute velocity and displacement vectors by integrating the base acceleration $\mathbf{a}_b$. These variables are available with the names `<phys>.v_abs` and `<phys>.u_abs` (for example, `solid.u_absX` for the X-component of the absolute displacement vector). For further details, see Frame Acceleration Forces in the *Structural Mechanics Theory* chapter

	A degree of freedom should either be constrained or free in the direction where the base excitation acts. Having a constraint acting in a local direction that is not either aligned or orthogonal to the excitation may produce unexpected results.

	As other global physics features, **Base Excitation** is global only within the interface, where it has been added. In case of several coupled structural interfaces in the model, you will need to add such feature under each of the involved interfaces. It will also have no effect on nonstructural interfaces involved in a multiphysics coupling such as, for example, fluid–structure interaction.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Base Excitation**
Shell > **Base Excitation**
Layered Shell > **Base Excitation**
Plate > **Base Excitation**
Membrane > **Base Excitation**
Beam > **Base Excitation**
Pipe Mechanics > **Base Excitation**
Truss > **Base Excitation**
Wire > **Base Excitation**
Multibody Dynamics > **Base Excitation**

Ribbon

Physics tab with a structural mechanics interface selected:

Global > **Base Excitation**

Inertia Relief

Inertia relief analysis provides a possibility to perform stationary analysis on an unconstrained structure. The assumption is that the structure is accelerated by external loads so that the inertial forces balances the applied load. This procedure requires that up to seven extra linear load cases are solved, in order to find the acceleration field.

The **Inertia Relief** node serves multiple purposes. It provides a set of built-in constraints to avoid singularity, sets up the equation system to be solved for finding force balance, and provides the automatic generation of necessary nodes in the Model Builder tree.

INERTIA RELIEF

In order to compute the inertial forces, the external loads are balanced so that there are no reaction forces at the *control points*. The control points are automatically constrained in a way that no rigid-body modes are present.

Choose a method for selecting the **Control points** — **Automatic** or **From selection input**. When using the automatic selection, the control points are chosen by the program. In

most cases, this works fine. There are, however, some cases in which certain points are not suitable as control points. Some examples are:

- A point load is applied to the point.
- There are internal constraints (for example from multiphysics couplings) at a point.

In such cases, you can use **From selection input**. Then, a **Point Selection** section is shown. Select the points to be used for constraining the domain.

- In 3D, select three points that are not collinear.
- In 2D, select two points.
- In 2D axisymmetry, select one point, not on the axis of rotation.

When the settings in this node are ready, click the **Automated Model Setup** icon ( ) on the **Inertia Relief** section toolbar and select the **Create Load Groups and Study** ( ) button to set up load groups for the load cases and a dedicated study.

- No boundary conditions that suppress rigid body motion can be present. This applies not only to constraints, but also to, for example, elastic foundations.
- If you have several structural mechanics interfaces in your model, you should add an **Inertia Relief** node to each of them. You should, however, only click the **Create Load Groups and Study** button in one of them.
- Since it is based on linear superposition of precomputed load cases, the inertia relief analysis can produce incorrect results in case of geometric nonlinearity.
- Do not manually edit the created load groups, or the settings for the load case handling in the study.
- Once you have run the special training study for inertia relief, any other study will contain the computed rigid body acceleration field. If the external loads are changed, then you need to re-run the training study.

For more information, see Inertia Relief in the *Structural Mechanics Theory* chapter.

Bracket — Inertia Relief Analysis: Application Library path
Structural_Mechanics_Module/Tutorials/bracket_inertia_relief

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Domain Constraints** > **Inertia Relief**
Shell > **Face Constraints** > **Inertia Relief**
Layered Shell > **Face and Domain Constraints** > **Inertia Relief**
Plate > **Face Constraints** > **Inertia Relief**
Membrane > **Face Constraints** > **Relief**
Beam > **Line Constraints** > **Inertia Relief**
Pipe Mechanics > **Line Constraints** > **Inertia Relief**
Truss > **Line Constraints** > **Inertia Relief**
Wire > **Line Constraints** > **Inertia Relief**
Multibody Dynamics > **Domain Constraints** > **Inertia Relief**

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Domains > **Domain Constraints** > **Inertia Relief**

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Face Constraints** > **Inertia Relief**

Physics tab with **Plate** selected:

Domains > **Face Constraints** > **Inertia Relief**

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Inertia Relief**

Physics tab with **Beam** or **Truss** selected:

Edges > **Line Constraints** > **Inertia Relief**

Rotating Frame

Centrifugal, Coriolis, and Euler forces are "fictitious" volume forces that need to be introduced in a rotating frame of reference, since it is not an inertial system. Use a **Rotating Frame** node to add the effect of these forces. The forces are generated by all

selected features in the physics interface having a density, mass, or mass distribution. You select objects having the highest geometrical dimension of the interface, and all objects with a lower dimensionality that belong to the selection are automatically included.

The idea when using a rotating frame is that the observer rotates with the structure. All results, such as displacements, are thus relative to the rotating frame.

The **Rotating Frame** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). **Rotating Frame** is available in 3D, 2D, and 2D axisymmetry.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Rotating Frame node.

ROTATING FRAME

Select an **Axis of rotation** — **x-axis**, **y-axis**, **z-axis**, or **User defined**. For **User defined** enter a **Rotation axis base point** $\mathbf{r}_{bp}$ and **Rotation axis direction** $\mathbf{e}_{ax}$.

Select a **Rotational velocity**— **Angular velocity**, **Revolutions per time**, **Rigid body**, or **User defined.**

The option **Rigid body** is only available in the Solid Mechanics and Multibody Dynamics interfaces.

- For **Angular velocity** enter a value for the angular velocity, ω_r.
- For **Revolutions per time**, enter a value for the rotational velocity in revolutions per time, Ω_r.
- For **Rigid body**, enter an **Applied moment**, M, and an **Initial angular velocity**, ω_0. In this case, the angular speed is variable over time, and computed by using time integration of the angular acceleration caused by the applied moment.

$$I_{\mathrm{ax}} \frac{\partial \omega_{\mathrm{r}}}{\partial t} = M$$

$$I_{\mathrm{ax}} = \int \rho r_{\mathrm{p}}^2 dV$$

It is assumed that axis of orientation is kept fixed. The moment of inertia around the axis of rotation, I_{ax}, is automatically computed from the mass properties if the selection.

- For **User defined**, enter the **Rotation angle** α in radians as function of time (the variable `t`).

Select a **Rotational direction** — **Counterclockwise** or **Clockwise**. The rotational direction does not make any difference for a centrifugal force. Therefore, this input is only available if Coriolis or Euler forces are included, or if the **Define spatial frame rotation** checkbox is selected.

The rotational direction can be specified either from the **Rotational direction** or from the sign of the **Angular velocity** or **Revolutions per time** inputs. That is entering a negative angular velocity will counteract the **Clockwise** option in **Rotational direction**.

In the case of a geometrically nonlinear analysis, the displacements computed in the Solid Mechanics interface are used to define the spatial frame. When a rotating frame is used, there are two possible interpretations.

- The default is that the spatial frame is also interpreted as corotating, that is only the displacements relative to the rotation are defining the spatial frame.
- Select **Define spatial frame rotation** to add also the rotation to the definition of the spatial frame. For other physics interfaces, this is similar to specifying a Rotating Domain, but with the displacements added. This approach must be used if there are a mix of stationary and rotating domains.

When **Define spatial frame rotation** is selected, the actual angle of rotation becomes important. If the angular speed is nonconstant, you must then enter an expression for the angle as function of time explicitly using the **User defined** option. For the two other input options, the angle of rotation is assumed to be $\alpha_0 + \omega_{\mathrm{r}} t$.

FRAME ACCELERATION EFFECT

For 3D and 2D components, use the **Centrifugal force**, **Coriolis force**, or **Euler force** checkboxes to determine which effects of a rotating frame that are to be incorporated in the analysis. Only **Centrifugal force** is selected by default.

For 2D axisymmetric components, the only effect from a rotating frame is the centrifugal force, which is then always included.

The **Spin softening** checkbox is selected by default. When including spin-softening effects, an extra contribution to the centrifugal force from deformation is taken into account. The **Spin softening** checkbox is only available if **Centrifugal force** or **Euler force** is selected.

For more information, see Frame Acceleration Forces in the *Structural Mechanics Theory* chapter.

You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Volume Forces** > **Rotating Frame**
Solid Mechanics, Explicit Dynamics > **Volume Forces** > **Rotating Frame**
Shell > **Face and Volume Loads** > **Rotating Frame**
Layered Shell > **Face and Volume Loads** > **Rotating Frame**
Plate > **Face and Volume Loads** > **Rotating Frame**
Membrane > **Face and Volume Loads** > **Rotating Frame**
Beam > **Line and Volume Loads** > **Rotating Frame**
Truss > **Line and Volume Loads** > **Rotating Frame**
Truss, Explicit Dynamics > **Line and Volume Loads** > **Rotating Frame**
Wire > **Line and Volume Loads** > **Rotating Frame**
Multibody Dynamics > **Volume Forces** > **Rotating Frame**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Domains > **Volume Forces** > **Rotating Frame**

Physics tab with **Shell**, **Layered Shell**, or **Membrane** selected:

Boundaries > **Face and Volume Loads** > **Rotating Frame**

Physics tab with **Plate** selected:

Domains > **Face and Volume Loads** > **Rotating Frame**

Physics tab with **Beam**; **Truss**; **Truss, Explicit Dynamics**; or **Wire** selected:

Edges > **Line and Volume Loads** > **Rotating Frame**

Linearly Accelerated Frame

When you add a **Linearly Accelerated Frame** node, inertial forces are applied to all selected features in the physics interface with a density, mass, or mass distribution. You select objects having the highest geometrical dimension of the interface, and all objects with a lower dimensionality that belong to the selection are automatically included.

The **Linearly Accelerated Frame** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Linearly Accelerated Frame node.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system must have orthonormal coordinate axes, and be defined in the spatial frame. Select a coordinate system in which the frame acceleration vector is prescribed.

LINEARLY ACCELERATED FRAME

Enter the components of the linear **Frame acceleration**, $\mathbf{a}_f$. The force is computed as $\mathbf{f} = -\rho\mathbf{a}_f$ and acts in a fixed spatial direction.

TIME-DEPENDENT STUDY

In a time-dependent study, you can compute the absolute velocity and absolute displacement postprocessing variables by checking the corresponding checkboxes. The variables for the absolute acceleration, velocity, and displacement are available with the

names `<phys>.<lafm_tag>.a_abs`, `<phys>.<lafm_tag>.v_abs`, and `<phys>.<lafm_tag>.u_abs` (for example, `solid.lafm1.u_absX` for the *X*-component of the absolute acceleration vector). Note that these variables will only take the frame acceleration specified in the current node into account, thus excluding contributions added by other nodes in the same physics interface, such as **Base Excitation**, **Gravity**, **Rotating Frame**, or another **Linearly Accelerated Frame**.

For more information, see Frame Acceleration Forces in the *Structural Mechanics Theory* chapter.

Only features that have a geometrical selection contribute to the mass forces. The **Mass and Moment of Inertia** nodes are global features and will not get any contribution from **Linearly Accelerated Frame**. You may need to add extra force contributions if such nodes are present.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed acceleration.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Volume Forces** > **Linearly Accelerated Frame**
Solid Mechanics, Explicit Dynamics > **Volume Forces** > **Linearly Accelerated Frame**
Shell > **Face and Volume Loads** > **Linearly Accelerated Frame**
Layered Shell > **Face and Volume Loads** > **Linearly Accelerated Frame**
Plate > **Face and Volume Loads** > **Linearly Accelerated Frame**
Membrane > **Face and Volume Loads** > **Linearly Accelerated Frame**
Beam > **Line and Volume Loads** > **Linearly Accelerated Frame**
Truss > **Line and Volume Loads** > **Linearly Accelerated Frame**
Truss, Explicit Dynamics > **Line and Volume Loads** > **Linearly Accelerated Frame**
Wire > **Line and Volume Loads** > **Linearly Accelerated Frame**
Multibody Dynamics > **Volume Forces** > **Linearly Accelerated Frame**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Domains > **Volume Forces** > **Linearly Accelerated Frame**

Physics tab with **Shell**, **Layered Shell**, or **Membrane** selected:

Boundaries > **Face and Volume Loads** > **Linearly Accelerated Frame**

Physics tab with **Plate** selected:

Domains > **Face and Volume Loads** > **Linearly Accelerated Frame**

Physics tab with **Beam**; **Truss**; **Truss, Explicit Dynamics**; or **Wire** selected:

Edges > **Line and Volume Loads** > **Linearly Accelerated Frame**

Boundary Load

Use a **Boundary Load** to apply tractions or pressure on boundaries.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Boundary Load node.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, **Pressure**, or **Resultant**. For 2D components it is possible to also select **Force per reference length** or **Force per deformed length**. Then enter values or expressions based on the selection and the space dimension.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.

- For **Total force**, COMSOL Multiphysics divides the total force by the area of the boundaries where the load is active. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Pressure**, a scalar input is given, and the orientation of the load is given by the normal to the boundary. The pressure is positive when directed toward the solid. In a geometrically nonlinear analysis, the current surface normal and area are used.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected boundaries. For **Point**, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

TABLE 4-5: BOUNDARY LOAD TYPES.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference area	$\mathbf{f}_A$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)
Force per deformed area	$\mathbf{f}_a$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)
Force per reference length	$\mathbf{f}_L$	N/m	boundaries	2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m	boundaries	2D (x, y)

TABLE 4-5: BOUNDARY LOAD TYPES.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Total force	$\mathbf{F}_{tot}$	N	boundaries	3D (x, y, z) 2D (x, y) 1D (x) 2D axisymmetric (r, z) 1D axisymmetric (r)
Pressure	p	Pa	boundaries	3D 2D 2D axisymmetric
Resultant	**F**, **M**	N,N·m	boundaries	3D (x, y, z) 2D (x, y)

TRACTION FIELD

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and `<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions.

When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{sym}$, and a **Point**, $\mathbf{x}_{sym}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Boundary Load**
Solid Mechanics > **Thin Layer** > **Boundary Load**
Solid Mechanics, Explicit Dynamics > **Boundary Load**
Layered Shell > **Boundary Load**
Multibody Dynamics > **Boundary Load**

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Solid Mechanics** > **Body Load**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **Solid Mechanics** > **Body Load**

Physics tab with **Multibody Dynamics** selected:

Boundaries > **Multibody Dynamics** > **Body Load**

Physics tab with **Layered Shell** selected:

Edges > **Layered Shell** > **Boundary Load**

Physics tab with **Thin Layer** selected:

Attributes > **Boundary Load**

Edge Load

Add an **Edge Load** to 3D components to apply a force distributed along an edge.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Edge Load node.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select a **Load type** — **Force per reference length**, **Force per deformed length**, or **Total force**. Then enter values or expressions for the components in based on the selection:

- The load per reference unit length $\mathbf{f}_{\mathrm{L}}$.
- The load per deformed unit length $\mathbf{f}_{\mathrm{l}}$.
- The total force $\mathbf{F}_{\mathrm{tot}}$. COMSOL Multiphysics then divides the total force by the length of the edge where the load is active, and then the edge load is applied in the same way as in **Force per reference length**.

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Edges** > **Edge Load**
Solid Mechanics, Explicit Dynamics > **Edges** > **Edge Load**
Layered Shell > **Edge Load**
Multibody Dynamics > **Edges** > **Edge Load**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Edges > **Edge Load**

Physics tab with **Layered Shell** selected:

Edges > **Layered Shell** > **Edge Load**

Point Load

Add a **Point Load** to points for concentrated forces at points in 2D and 3D.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Point Load node.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select the **Load type** — **Force per point**, **Total force**, or **Resultant**. Then enter values or expressions based on the selection.

- For **Force per point** enter values or expressions for the components of the point load $\mathbf{F}_p$.
- For **Total force**, COMSOL Multiphysics divides the total force by the number of points where the load is active. Then the force is applied in the same way as for **Force per point**.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected points. For **Point**, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

The **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

TRACTION FIELD

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and `<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Points** > **Point Load**
Solid Mechanics, Explicit Dynamics > **Points** > **Point Load**
Layered Shell > **Points** > **Interfaces** > **Point Load**
Truss > **Point Load**
Truss, Explicit Dynamics > **Point Load**
Wire > **Point Load**
Multibody Dynamics > **Points** > **Point Load**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Points > **Point Load**

Physics tab with **Layered Shell** selected:

Points > **Interfaces** > **Point Load**

Physics tab with **Truss** selected:

Points > **Truss** > **Point Load**

Physics tab with **Truss, Explicit Dynamics** selected:

Points > **Truss, Explicit Dynamics** > **Point Load**

Physics tab with **Wire** selected:

Points > **Wire** > **Point Load**

Point Load, Free

Use a **Point Load, Free** node to add concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Imported meshes, where there may not be suitable points for load application

- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free point load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable *<phys>*.*<load_tag>*.num_ignored (for example, solid.plf1.num_ignored).
- The local stress field in the element where a point load acts will not be reliable.

The **Point Load, Free** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available in 3D, 2D, and 2D axisymmetry.

LOCATION AND FORCE

Select a **Frame** — **Material** or **Spatial**, which determines how the given coordinates are interpreted in case of a geometrically nonlinear analysis. When **Material** is selected, the load location on the body is interpreted with respect to the initial configuration. When **Spatial** is selected, then the load location is given in space fixed coordinates, which usually means that the material point on the body where the load is applied will change with deformation even if the coordinate values are constant.

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**X**, **Y**, **Z**). If spatial frame input is used, then the spatial coordinates (**x**, **y**, **z**) are used. Then, enter the force values, **FxI**, **FyI**, and **FzI**. The force vector is interpreted in the coordinate system selected in the **Coordinate System Selection** section.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

Pratt Truss Bridge: Application Library path
Structural_Mechanics_Module/Beams_and_Shells/pratt_truss_bridge

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Point Load, Free**
Membrane > **Point Load, Free**
Truss > **Point Load, Free**
Wire > **Point Load, Free**
Multibody Dynamics > **Point Load, Free**

Ribbon

Physics tab with **Solid Mechanics** selected:

Global > **Solid Mechanics** > **Point Load, Free**

Physics tab with **Membrane** selected:

Global > **Membrane** > **Point Load, Free**

Physics tab with **Truss** or **Wire** selected:

Global > **Truss** > **Point Load, Free**

Physics tab with **Multibody Dynamics** selected:

Global > **Multibody Dynamics** > **Point Load, Free**

Ring Load

A load applied to a point, not located at the axis of revolution, in a 2D axisymmetric model actually represents a ring load. Add a **Ring Load** to points located at $R > 0$ to model such loads.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select the **Load type** — **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for $\mathbf{f}_L$, $\mathbf{f}_l$, or $\mathbf{F}_{tot}$.

- The **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.
- The **Total force** option should not be directly interpreted as a resultant, since the orientation is changing. Rather, it should be interpreted as the per reference length values, multiplied by $2\pi r$, where r is the radius of the point.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Points** > **Ring Load**

Membrane > **Ring Load**

Ribbon

Physics tab with **Solid Mechanics** selected:

Points > **Ring Load**

Physics tab with **Membrane** selected:

Points > **Membrane** > **Ring Load**

Ring Load, Free

A load applied to a point, not located at the axis of revolution, in a 2D axisymmetric model actually represents a ring load. Add a **Ring Load, Free** node to describe concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Imported meshes, where there may not be suitable points for load application

- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable *<phys>*.*<load_tag>*.num_ignored (for example, solid.rlf1.num_ignored).
- The local stress field in the element where a point load acts will not be reliable.

LOCATION AND FORCE

Select a **Frame** — **Material** or **Spatial** that determines how the given coordinates are interpreted in case of a geometrically nonlinear analysis. When **Material** is selected, the load location on the body is interpreted with respect to the initial configuration. When **Spatial** is selected, then the load location is given in space fixed coordinates, which usually means that the material point on the body where the load is applied will change with deformation even if the coordinate values are constant.

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**R**, **Z**). If spatial frame input is used, then the spatial coordinates (**r**, **z**) are used. Then, enter the force values, **FxI** and **FyI**. The force vector is interpreted in the coordinate system selected in the **Coordinate System Selection** section.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Ring Load, Free**

Membrane > **Ring Load, Free**

Ribbon

Physics tab with **Solid Mechanics** selected:

Global > **Solid Mechanics** > **Ring Load, Free**

Physics tab with **Membrane** selected:

Global > **Membrane** > **Ring Load, Free**

Point Load (on Axis)

A **Point Load (on Axis)** node can be added to points located at $R = 0$ in 2D axially symmetric models. This is the only true point load an axisymmetric model, since loads applied at points having nonzero radial coordinates actually represent a ring load. You can add a ring load using a **Ring Load** node.

FORCE

Enter values or expressions for the **Force** F_z in the axial direction.

The **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Points** > **Point Load (on Axis)**
Solid Mechanics, Explicit Dynamics > **Points** > **Point Load (on Axis)**
Shell > **Point Load (on Axis)**
Membrane > **Point Load (on Axis)**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Points > **Point Load (on Axis)**

Physics tab with **Shell** or **Membrane** selected:

Points > **Shell** > **Point Load (on Axis)**
Points > **Membrane** > **Point Load (on Axis)**

Thin Layer

Use the **Thin Layer** node to apply elastic and damping conditions between two parts, either on an interior boundary or between two boundaries forming a pair, or a thin cladding on exterior boundaries.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to apply this condition. An identity pair has to be created first.

BOUNDARY PROPERTIES

Select a **Thin layer type** — **Nonlayered** or **Layered**. This setting is not applicable when the **Thin Layer** node is selected from the **Pairs** menu.

- For **Nonlayered** (the default), enter the thin layer **Thickness** L_{th}.
- For **Layered**, and when **Use all layers** is selected, all information is taken from the definitions made in the **Single Layer Material**, **Layered Material Link**, or **Layered Material Stack** nodes under the **Materials** node in the current component.
- For **Layered**, and when **Use all layers** is not selected, the **Layered material** list is displayed. You can select from the current component:

1. Any **Single Layer Material** available under the **Materials** node.
2. Any **Layered Material Link** available under the **Materials** node.
3. Any **Layered Material Stack** available under the **Materials** node.

In the *COMSOL Multiphysics Reference Manual:* Single Layer Materials, Layered Material Link, and Layered Material Stack

If this node is selected from the **Pairs** menu, select the **Thickness definition** — **From geometry** or **User defined**.

- For **User defined** (the default), enter the thin layer **Thickness** L_{th}.
- For **From geometry**, the thin layer thickness is calculated from the actual gap distance in the geometry. This setting can only be used if there is a nonzero gap across the entire pair selection. The gap distance and the curvature of the boundaries are expected to be small.

THIN LAYER

Select the thin layer **Approximation** — **Solid**, **Membrane**, or **Spring**. Only the **Solid** approximation is available for **Layered** structures, and only the **Solid** or **Spring** approximations are available when the thin layer is selected from the **Pairs** menu.

By adding the following subnodes to the **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** nodes you can incorporate many effects in thin layers:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Polymer Viscoplasticity
- Fiber (Thin Layer)
- Damage
- Mullins Effect

When the **Spring** approximation is chosen, it is possible to specify a Spring Material for the thin layer.

The **Thin Layer** node and subnodes are only available with some COMSOL products (see www.comsol.com/products/specifications/).

The **Thin Layer** node is available in 3D, 2D, and 2D axisymmetry.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer**

Solid Mechanics > **Pairs** > **Thin Layer**

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Thin Layer**

Pairs > **Thin Layer**

Fiber (Thin Layer)

Add one or more **Fiber** nodes to add stiffness in specified directions to a material in a thin layer. The **Fiber** feature can be used together with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** when the **Thin layer type** is **Nonlayered**. The assumption is that the volume of the fibers is small when compared to the base material.

You can include the effect of thermal expansion of the fiber by adding a **Thermal Expansion** subnode (see Thermal Expansion (Fiber)).

FIBER MODEL

Linear Elastic Material or Nonlinear Elastic Material

Select a **Material Model** — **Linear elastic** or **Uniaxial data**.

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- For **Linear elastic**, enter the **Young's modulus** E_{fiber} and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions.
- For **Uniaxial data**, enter the **Uniaxial stress function** $\sigma_{\text{ax,a}}$ and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions. The default user defined expression for $\sigma_{\text{ax,a}}$ is the linear function `210[GPa]*<item>.eea`, where the variable `<item>.eea` is the elastic strain in the fiber direction, and `<item>` corresponds to the tag of the **Fiber** node, for example

`solid.tl1.lemm1.fibt1`. See Fibers for Linear and Nonlinear Elastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

Hyperelastic Material

Select a **Material Model** — **Holzapfel–Gasser–Ogden**, **Linear elastic**, **Uniaxial data**, or **User defined**.

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- The **Holzapfel–Gasser–Ogden** model defines an exponential strain energy density W_{fiber}. The values for the **Fiber stiffness** k_1, **Model parameter** k_2, and **Fiber dispersion** k_3 are as a default taken **From material**. Select **User defined** to enter other values or expressions.
- Select **Linear elastic** or **Uniaxial data** to describe the fiber stiffness. See Linear Elastic Material or Nonlinear Elastic Material for details.
- If **User defined** is selected from the list, select a **Material symmetry** — **Isotropic**, **Transversely isotropic**, or **Anisotropic**.

  Depending on the choice, different stretch invariants are available to define the **Fiber strain energy density** W_{fiber}. The default expression for W_{fiber} is the function `0[Pa]*(<item>.IaCIe-1)`, where the variable `<item>.IaCIe` is the isochoric stretch invariant in the fiber direction, and `<item>` corresponds to the tags of the **Fiber** node, for example `solid.tl1.hmm1.fibt1`. See Fibers for Hyperelastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

For the **Holzapfel–Gasser–Ogden** model, select the **Use isochoric deformation** checkbox to account for the fiber incompressibility. See Compressible Fibers in the *Structural Mechanics Theory* chapter for details.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

DISTRIBUTION AND ORIENTATION

From the **Specify** list select how the fiber volume fraction is entered — **Volume distribution** or **Surface distribution**.

- For **Volume distribution**, enter the fiber **Volume fraction** v_{fiber}. To be consistent with the underlying assumptions, it should not exceed a few percent.
- For **Surface distribution**, the fibers are placed at the midsurface. Enter the fiber **Diameter** d_{fiber} and the fiber **Spacing** s_{fiber} to compute the fiber volume fraction. To be consistent with the underlying assumptions, the diameter–spacing ratio should not exceed a few percent.

Select a **Fiber orientation** from the list. The available choices are the axis directions of the coordinate system selected in the **Coordinate System Selection** section, or **User defined**. For **User defined**, enter the **Direction**. The direction vector $\mathbf{a}$ is interpreted in the in-plane directions of the selected local coordinate system.

See also Distributed Fiber Models in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Linear Elastic Material** > **Fiber**
Solid Mechanics > **Thin Layer** > **Nonlinear Elastic Material** > **Fiber**
Solid Mechanics > **Thin Layer** > **Hyperelastic Material** > **Fiber**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected under a **Thin Layer** node in the model tree:

Attributes > **Fiber**

Prescribed Displacement (Thin Layer)

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in one or more directions to the boundary.

If a displacement is prescribed in only one direction, this leaves the thin layer free to deform in the other directions.

You can also define more general displacements as a linear combination of the displacements in each direction, see Prescribed Displacement.

If a zero displacement is applied in all directions, this is the same as a Fixed Constraint (Thin Layer).

For details, see Prescribed Displacements, Velocities, and Accelerations.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed displacements and rotations are specified along the axes of this coordinate system.

PRESCRIBED DISPLACEMENT

For the displacement in each direction, select a setting from the list — **Free**, **Prescribed**, or **Limited**. Select:

- **Free** (the default) to leave the displacement component unconstrained.
- **Prescribed** to constrain the displacement component to a given value. Enter a scalar value for the component of the prescribed displacement u_0.
- **Limited** to set a maximum and a minimum limit for the displacement component. Enter values for the maximum displacement $u_{0,\max}$ and the minimum displacement $u_{0,\min}$. By default, they are set to `Inf` and `-Inf`, which corresponds to no active constraint.

If any displacement component is set to **Limited**, an additional section **Limited Displacement** is visible. Select the **Method** used to implement the weak inequality constraint — **Penalty** or **Augmented Lagrangian**. For both methods, enter a **Penalty factor** k_{p}.

By default, the **Penalty** method is suggested, which in principle enforces the maximum and minimum limits for the displacement by adding nonlinear springs with a stiffness equal to k_p when the limits are exceeded. This method is usually robust, but the accuracy is directly dependent on the chosen penalty factor.

The **Augmented Lagrangian** method adds extra degrees of freedom to improve the accuracy of the constraint. Here, the penalty factor is a numerical parameter, and has less impact on the accuracy of the constraint compared to when using the penalty method. The implementation of the augmented Lagrangian method puts no restrictions on the solver sequence, but for good convergence, proper scaling of the extra degrees of freedom can be important.

The default value for the **Penalty factor** k_p on what type of entity the **Prescribed Displacement** node is added. In the expressions below, `<phys>` is the tag of the physics interface, and `<pd>` is the tag of the **Prescribed Displacement** node.

- For edges, the default expression is `100*<phys>.Eequ`
- For boundaries, the default expression is `100*<phys>.Eequ/<phys>.<pd>.charLen`

The variable `<phys>.<pd>.charLen` is by default equal to the length of the mesh element.

For details about setting maximum and minimum limits for the displacements, see Limited Displacement

In versions prior to 6.2, you could chose to formulate the constraint in two ways: standard notation or general notation. Since it was rarely used, the general notation option is no longer supported. There are more natural way of specifying identical constraints.

If you have a model created in version 6.1 or earlier, where general notation was used, the settings for this option are visible, and are still functional. You should, however, consider to reformulate the constraints, since the general notation may no longer be supported in future versions.

For reference, the documentation for the general notation is provided below:

You can specify the displacements using a general notation that includes any linear combination of displacement components. For example, for 2D components, use the relationship

$$H\begin{bmatrix} u \\ v \end{bmatrix} = R$$

For **H** matrix H select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and then enter values as needed in the field or matrix. Enter values or expressions for the **R** vector R.

For example, to achieve the condition $u = v$, use the settings

$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which force the domain to move only diagonally in the xy-plane.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about this section, see Constraint Settings in the *COMSOL Multiphysics Reference Manual*.

EXCLUDED POINTS

To display these sections, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about these

sections, see Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Prescribed Displacement** (Edge)
Solid Mechanics > **Thin Layer** > **Points** > **Prescribed Displacement**

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Prescribed Displacement** (Edge)
Attributes > **Points** > **Prescribed Displacement**

Fixed Constraint (Thin Layer)

The **Fixed Constraint** and Prescribed Displacement (Thin Layer) nodes are similar, in that the prescribed displacement of the structural part on which it is acting is prescribed to zero. The displacements and rotations are prescribed to zero in all directions. There are no settings for this node. See Fixed Constraint.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about this section, see Constraint Settings in the *COMSOL Multiphysics Reference Manual*.

EXCLUDED POINTS

To display these sections, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about these sections, see Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Fixed Constraint** (Edge)
Solid Mechanics > **Thin Layer** > **Points** > **Fixed Constraint**

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Fixed Constraint** (Edge)
Attributes > **Points** > **Fixed Constraint**

Roller (Thin Layer)

The **Roller** node adds constraints on the edges of the **Thin Layer** node. The displacement is zero in the normal direction, but the thin layer is free to move in the other directions. See Roller.

FACE DEFINING THE LOCAL ORIENTATIONS

When an edge is shared between two or more boundaries, the local directions may not always be unique. It is then possible to use the control **Face Defining the Local Orientations** to select from which boundary the normal direction should be picked. The default is **Use face with lowest number**.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about this section, see Constraint Settings in the *COMSOL Multiphysics Reference Manual*.

EXCLUDED POINTS

To display these sections (**Excluded Edges** in 3D geometries only), click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about these sections, see Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Roller** (Edge)

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Roller**

Face Load (Thin Layer)

Add a **Face Load** node to apply tractions or pressure on boundaries where the **Thin Layer** material is active.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system.

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, **Pressure**, or **Resultant**. For 2D components it is possible to also select **Force per reference length** or **Force per deformed length**. Then enter values or expressions based on the selection and the space dimension.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.
- For **Total force**, COMSOL Multiphysics divides the total force by the area of the boundaries where the load is active. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Pressure**, a scalar input is given, and the orientation of the load is given by the normal to the boundary. The pressure is positive when directed toward the solid. In a geometrically nonlinear analysis, the current surface normal and area are used.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected boundaries. For

Point, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference area	$\mathbf{f}_A$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per deformed area	$\mathbf{f}_a$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per reference length	$\mathbf{f}_L$	N/m	boundaries	2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m	boundaries	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Pressure	p	Pa	boundaries	3D 2D 2D axisymmetric
Resultant	$\mathbf{F}$, $\mathbf{M}$	N, N·m	boundaries	3D (x, y, z) 2D (x, y)

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and

`<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Face Load**

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Face Load**

Boundary Load (Thin Layer)

Add a **Boundary Load** node to apply tractions or pressure on edges adjacent to boundaries where the **Thin Layer** node is active.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system.

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, **Pressure**, or **Resultant**. For 2D components it is possible to also select **Force per reference length** or **Force per deformed length**. Then enter values or expressions based on the selection and the space dimension.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.
- For **Total force**, COMSOL Multiphysics divides the total force by the area of the boundaries where the load is active. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected boundaries. For

Point, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference area	$\mathbf{f}_A$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per deformed area	$\mathbf{f}_a$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per reference length	$\mathbf{f}_L$	N/m	boundaries	2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m	boundaries	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Resultant	$\mathbf{F}$, $\mathbf{M}$	N, N·m	boundaries	3D (x, y, z) 2D (x, y)

TRACTION FIELD

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and

`<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Boundary Load**

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Boundary Load**

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

COORDINATE SYSTEM SELECTION

The spring constants and loss factors are given with respect to the selected coordinate system.

SPRING

Select the **Spring type** and its associated spring constant or force using Table 4-6 as a guide.

Use the built-in variables for the thin layer extension for the options **Force per area as function of extension**, **Force per length as function of extension**, or **Total force as function of extension**. The names are `<item>.uelt1`, `<item>.uelt2`, and `<item>.ueln`, where `<item>` is the name of the Thin Layer node (for example, `solid.tl1`), and `t1`, `t2`, and `n` are the coordinate names of the local boundary system.

The spring constants can be entered as **Isotropic**, **Diagonal**, or **Symmetric**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

TABLE 4-6: SPRING TYPES FOR THE SPRING MATERIAL FEATURE.

SPRING TYPE	VARIABLE	SI UNITS	SPACE DIMENSION
Spring constant per reference area	$\mathbf{k}_A$	$N/(m \cdot m^2)$	3D, 2D, and 2D axisymmetric
Spring constant per reference length	$\mathbf{k}_L$	$N/(m \cdot m)$	2D
Total spring constant	$\mathbf{k}_{tot}$	N/m	3D, 2D, and 2D axisymmetric
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	3D, 2D, and 2D axisymmetric
Force per length as function of extension	$\mathbf{F}_L$	N/m	2D
Total force as function of extension	$\mathbf{F}_{tot}$	N	3D, 2D, and 2D axisymmetric

Enter the **Mass type** of the thin elastic layer — **Total mass**, **Mass per reference volume**, ρ_V, or **Mass per reference area** ρ_A. For thin layers on 2D geometries, it is also possible to enter the **Mass per reference length** ρ_L.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about this section, see Constraint Settings in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Spring Material**

Ribbon

Physics tab with **Thin Layer** node selected in the model tree:

Attributes > **Spring Material**

Damping (Spring Material)

From the **Damping type** list, select **Loss factor**, **Viscous damping**, **Damping constant per reference area**, or **Total damping constant**. For thin layers on 2D geometries, it is also possible to enter the **Damping constant per reference length** $\mathbf{d}_{\mathrm{L}}$.

When choosing **Loss factor**, from the **Loss factor type** list select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_{s}, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, or **Symmetric**, then enter values or expressions for the **Loss factor for spring** η_{k}. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

When choosing **Viscous damping**, enter the **Normal viscosity** η_b and the **Shear viscosity** η_v.

When choosing **Damping constant per reference area**, select **Isotropic**, **Diagonal**, or **Symmetric**, then enter values or expressions for the **Damping constant per reference area** $\mathbf{d}_A$.

When choosing **Damping constant per reference length**, select **Isotropic**, **Diagonal**, or **Symmetric**, then enter values or expressions for the **Damping constant per reference length** $\mathbf{d}_L$.

When choosing **Total damping constant**, select **Isotropic**, **Diagonal**, or **Symmetric**, then enter values or expressions for the **Total damping constant** $\mathbf{d}_{tot}$.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Thin Layer** > **Spring Material** > **Damping**

Ribbon

Physics tab with **Spring Material** node selected in the model tree:

Attributes > **Damping**

Spring Foundation

Use the **Spring Foundation** node to apply elastic and damping boundary conditions for domains, boundaries, edges, and points.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

The **Spring Foundation** and Thin Layer nodes are similar, with the difference that a **Spring Foundation** connects the structural part on which it is acting to a fixed "ground", while a **Thin Layer** acts between two parts, either on an interior boundary or between two boundaries forming a pair.

The **Spring Foundation** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Spring Foundation node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Spring Foundation, Interface node.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to apply this condition. An identity pair has to be created first.

COORDINATE SYSTEM SELECTION

The spring and damping constants are given with respect to the selected coordinate system.

SPRING

Select the **Spring type** and its associated spring constant or force using Table 4-7 as a guide. The default option is the spring type for the type of geometric entity and space dimension, and there are different combinations available based on this.

When the option is of the type 'force as function of extension', then the built-in variables describing the spring extension must be used in the expression as described in Springs and Dampers. The spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Use material data** is selected as **Spring type**, the spring stiffness values are computed from the given material data and a layer thickness. From the **Specify** list, select a pair of elastic properties — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, or **Bulk modulus and shear modulus**. Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another according to Table 4-8. For the chosen properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

Enter a **Thickness**, d_s, to specify the physical thickness of the elastic layer.

TABLE 4-7: SPRING TYPES FOR THE SPRING FOUNDATION FEATURE.

SPRING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Spring constant per reference volume	$\mathbf{k}_V$	N/(m·m^3)	domains	3D, 2D, 1D, and 2D/1D axisymmetric
Total spring constant	$\mathbf{k}_{tot}$	N/m	domains, edges	3D, 2D, 1D, and 2D/1D axisymmetric
Spring constant per reference area	$\mathbf{k}_A$	N/(m·m^2)	boundaries	3D, 2D
Spring constant per reference length	$\mathbf{k}_L$	N/(m·m)	edges, boundaries (2D)	3D, 2D, 1D
Spring constant	$\mathbf{k}_P$	N/m	points	3D, 2D, 1D, and 2D axisymmetric
Force per volume as function of extension	$\mathbf{F}_V$	N/m^3	domains	3D, 2D, 1D, and 2D/1D axisymmetric
Total force as function of extension	$\mathbf{F}_{tot}$	N	domains, boundaries, edges	3D, 2D, 1D, and 2D/1D axisymmetric
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	domains, boundaries	3D, 2D
Force per length as function of extension	$\mathbf{F}_L$	N/m	edges	3D/1D
Force as function of extension	$\mathbf{F}_P$	N	points	3D, 2D, 1D, and 2D/1D axisymmetric
Use material data	N/A	N/A	boundaries	3D, 2D, and 2D axisymmetric

TABLE 4-8: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$
Young's modulus	E	E	E	$\frac{9KG}{3K+G}$
Poisson's ratio	ν	ν	$\frac{E}{2G}-1$	$\frac{1}{2}\left(1-\frac{3G}{3K+G}\right)$
Bulk modulus	K	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	K
Shear modulus	G	$\frac{E}{2(1+\nu)}$	G	G

ROTATIONAL SPRING

This section is available for the Beam and Pipe Mechanics interfaces. All settings are analogous to the corresponding settings in the Spring section, but with forces replaced by moments and displacements replaced by rotations.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**. The latter option is not available for 1D or 1D axisymmetry.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

This section is available for the Beam and Pipe Mechanics interfaces. All settings are analogous to the corresponding settings in the Loss Factor Damping section.

VISCOUS DAMPING

Select the **Damping type** using Table 4-9 as a guide. The default option is the default damping type for the type of geometric entity and space dimension, and there are different combinations available based on this. The damping matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 4-9: DAMPING TYPES FOR THE SPRING FOUNDATION FEATURE.

DAMPING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Damping constant per reference volume	$\mathbf{d}_V$	$N·s/(m·m^3)$	domains, boundaries (2D)	3D, 2D, 1D
Damping constant per reference area	$\mathbf{d}_A$	$N·s/(m·m^2)$	domains, boundaries	3D, 2D, and 2D axisymmetric
Total damping constant	$\mathbf{d}_{tot}$	N·s/m	domains, boundaries, edges, points	3D, 2D, 1D, and 2D/1D axisymmetric
Damping constant per reference length	$\mathbf{d}_L$	N·s/(m·m)	edges	3D, 1D

ROTATIONAL VISCOUS DAMPING

This section is available for the Beam and Pipe Mechanics interfaces. All settings in this section are analogous to the corresponding settings in the Viscous Damping section, but with forces replaced by moments and velocities replaced by angular velocities.

ENERGY DISSIPATION

You can select to compute and store energy dissipation variables in a time-dependent analysis. Doing so will add extra degrees of freedom to the model.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by damping.

- Dissipated Energy
- Energy Variables

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

	You can activate and deactivate **Spring Foundation** by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

	• Springs and Dampers in the *Structural Mechanics Modeling* chapter. • Spring Foundation and Thin Elastic Layer in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Mass, Spring, and Damper** > **Spring Foundation** (Domain, Boundary)
Solid Mechanics > **Edges** > **Spring Foundation**
Solid Mechanics > **Points** > **Spring Foundation**
Solid Mechanics, Explicit Dynamics > **Mass, Spring, and Damper** > **Spring Foundation** (Domain, Boundary)
Solid Mechanics, Explicit Dynamics > **Edges** > **Spring Foundation**
Solid Mechanics, Explicit Dynamics > **Points** > **Spring Foundation**
Shell > **Mass, Spring, and Damper** > **Spring Foundation** (Boundary, Edge)
Shell > **Points** > **Spring Foundation**
Plate > **Mass, Spring, and Damper** > **Spring Foundation** (Domain, Boundary)
Plate > **Points** > **Spring Foundation**
Layered Shell > **Mass, Spring, and Damper** > **Spring Foundation** (Boundary, Edge)
Layered Shell > **Mass, Spring, and Damper** > **Interfaces** > **Spring Foundation, Interface** (Boundary, Edge)
Layered Shell > **Points** > **Spring Foundation**
Layered Shell > **Points** > **Interfaces** > **Spring Foundation, Interface**
Membrane > **Mass, Spring, and Damper** > **Spring Foundation** (Boundary, Edge)
Membrane > **Points** > **Spring Foundation**
Beam > **Mass, Spring, and Damper** > **Spring Foundation**
Pipe Mechanics > **Mass, Spring, and Damper** > **Spring Foundation**
Truss > **Mass, Spring, and Damper** > **Spring Foundation**
Truss, Explicit Dynamics > **Mass, Spring, and Damper** > **Spring Foundation**

Multibody Dynamics > **Mass, Spring, and Damper** > **Spring Foundation** (Domain, Boundary)
Multibody Dynamics > **Edges** > **Spring Foundation**
Multibody Dynamics > **Points** > **Spring Foundation**

Ribbon
Physics tab with **Solid Mechanics** selected:

Domains > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Solid Mechanics** > **Spring Foundation**
Points > **Solid Mechanics** > **Spring Foundation**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Solid Mechanics, Explicit Dynamics** > **Spring Foundation**
Points > **Solid Mechanics, Explicit Dynamics** > **Spring Foundation**

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Mass, Spring, and Damper** > **Spring Foundation**
Points > **Shell** > **Spring Foundation**
Points > **Membrane** > **Spring Foundation**

Physics tab with **Plate** selected:

Domains > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Points > **Plate** > **Spring Foundation**

Physics tab with **Layered Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Interfaces** > **Spring Foundation, Interface**
Edges > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Interfaces** > **Spring Foundation, Interface**
Points > **Layered Shell** > **Spring Foundation**
Points > **Interfaces** > **Spring Foundation, Interface**

Physics tab with **Beam**, **Pipe Mechanics**, **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Mass, Spring, and Damper** > **Spring Foundation**

Points > **Mass, Spring, and Damper** > **Spring Foundation**

Physics tab with **Multibody Dynamics** selected:

Domains > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Multibody Dynamics** > **Spring Foundation**
Points > **Multibody Dynamics** > **Spring Foundation**

Thin Elastic Layer

From version 6.2, the **Thin Elastic Layer** feature can no longer be added to a model. It is superseded by Thin Layer, which is more general. In models that were created in earlier versions, **Thin Elastic Layer** nodes can still be present and are fully functional.

You are, however, encouraged to replace any **Thin Elastic Layer** nodes with corresponding **Thin Layer** nodes.

Use the **Thin Elastic Layer** node to apply elastic and damping conditions between two parts, either on an interior boundary or on a boundary pair.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

The **Thin Elastic Layer** and Spring Foundation nodes are similar, with the difference that a Spring Foundation connects the structural part on which it is acting to a fixed "ground".

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to apply this condition. An identity pair has to be created first.

COORDINATE SYSTEM SELECTION

The spring and damping constants are given with respect to the selected coordinate system, with the exception that when the stiffness is specified by **Use material data**, the coordinate system selection is not used.

SPRING

Select the **Spring type** and its associated spring constant or force using Table 4-10 as a guide. The default option is the spring type for the type of geometric entity and space dimension, and there are different combinations available based on this.

When the option is of the type "force as function of extension", then the built-in variables describing the spring extension must be used in the expression as described in Springs and Dampers. The spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Use material data** is selected as **Spring type**, the spring stiffness values are computed from material data and layer thickness. From the **Specify** list, select a pair of elastic properties — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, or **Bulk modulus and shear modulus**. Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another according to Table 4-8. For the chosen properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

Enter a **Thickness**, d_s, to specify the physical thickness of the elastic layer.

TABLE 4-10: SPRING TYPES FOR THE THIN ELASTIC LAYER FEATURE.

SPRING TYPE	VARIABLE	SI UNITS	SPACE DIMENSION
Total spring constant	$\mathbf{k}_{tot}$	N/m	3D, 2D, and 2D axisymmetric
Spring constant per unit area	$\mathbf{k}_A$	N/(m·m^2)	3D, 2D, and 2D axisymmetric
Spring constant per unit length	$\mathbf{k}_L$	N/(m·m)	2D
Total force as function of extension	$\mathbf{F}_{tot}$	N	3D, 2D, and 2D axisymmetric
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	3D, 2D, and 2D axisymmetric
Force per length as function of extension	$\mathbf{F}_L$	N/m	2D
Use material data	N/A	N/A	3D, 2D, and 2D axisymmetric

ADDED MASS

Enter the added mass of the thin elastic layer. The type of input depends on the selected **Spring type**; see Table 4-10.

- For **Total spring constant** and **Total force as function of extension**, enter the **Total mass** m of the thin elastic layer.
- For **Spring constant per unit area** and **Force per area as function of extension**, enter the **Mass per unit area** ρ_A of the thin elastic layer.
- For **Spring constant per unit area** and **Force per area as function of extension**, enter the **Mass per unit length** ρ_L of the thin elastic layer.
- For **Use material data**, enter the **Density** ρ of the thin elastic layer. Either use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS DAMPING

Select the **Damping type** using Table 4-11 as a guide. The default option is the default damping type for the space dimension. The damping matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 4-11: DAMPING TYPES FOR THE THIN ELASTIC LAYER FEATURE.

DAMPING TYPE	VARIABLE	SI UNITS	SPACE DIMENSION
Damping constant per unit area	$\mathbf{d}_A$	N·s/(m·m^2)	3D, 2D, and 2D axisymmetric
Total damping constant	$\mathbf{d}_{tot}$	N·s/m	3D, 2D, and 2D axisymmetric
Damping constant per unit length	$\mathbf{d}_L$	N·s/(m·m)	2D

ENERGY DISSIPATION

You can select to compute and store energy dissipation variables in a time-dependent analysis. Doing so will add extra degrees of freedom to the model.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by damping.

- Dissipated Energy
- Energy Variables

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

- Springs and Dampers in the *Structural Mechanics Modeling* chapter.
- Spring Foundation and Thin Elastic Layer in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Mass, Spring, and Damper** > **Thin Elastic Layer** (Edge)

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Mass, Spring, and Damper** > **Thin Elastic Layer**
Pairs > **Solid Mechanics** > **Thin Elastic Layer**

Predeformation

Use the **Predeformation** subnode to specify that the elastic forces in the Spring Foundation or Spring Material are nonzero at zero displacement. Thus, you can model cases where the unstressed state of the spring is in another configuration than the one described by the geometry.

The value of the predeformation can vary during the simulation, so it should not be interpreted as an initial value.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Predeformation node.

SPRING PREDEFORMATION

Based on space dimension, enter the values for the **Spring Predeformation** $\mathbf{u}_0$.

You can assign the load caused by the predeformation to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Spring Foundation** > **Predeformation**
Solid Mechanics > **Thin Layer** > **Spring Material** > **Predeformation**
Solid Mechanics, Explicit Dynamics > **Spring Foundation** > **Predeformation**
Shell > **Spring Foundation** > **Predeformation**
Layered Shell > **Spring Foundation** > **Predeformation**
Layered Shell > **Spring Foundation, Interface** > **Predeformation**
Layered Shell > **Thin Elastic Layer** > **Predeformation**
Plate > **Spring Foundation** > **Predeformation**
Membrane > **Spring Foundation** > **Predeformation**
Beam > **Spring Foundation** > **Predeformation**
Truss > **Spring Foundation** > **Predeformation**
Truss, Explicit Dynamics > **Spring Foundation** > **Predeformation**
Wire > **Spring Foundation** > **Predeformation**
Multibody Dynamics > **Spring Foundation** > **Predeformation**

Ribbon

Physics tab with **Spring Foundation**, **Spring Foundation, Interface**, or **Spring Material** node selected in the model tree:

Attributes > **Predeformation**

Spring–Damper

Use a **Spring–Damper** ( ) for 3D, 2D, and 1D components to model — between two points — an elastic spring, a viscous damper, or both.

The spring can either act as an axial spring between the two points, or be defined be a general spring matrix, connecting all the degrees of freedom in the two points. The two points are referred to as the *source* and *destination* point, respectively. The points can be geometrical points, but there are also other methods for describing the attachment point of the spring as a virtual point in space.

A spring–damper can connect two points belonging to two different physics interfaces. If at least one of the physics interfaces possesses rotational degrees of freedom, the **Spring–Damper** should be placed in such an interface.

The **Spring–Damper** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

This section is only of importance when a full matrix representation of the spring–damper is used. In that case, the matrix is interpreted as acting on the degrees of freedom in the local directions provided by the local coordinate system.

SKETCH

This section is only available if **Directional** is selected as **Spring type**. In that case, a conceptual representation of the spring–damper assembly is shown.

ATTACHMENT SELECTION

In this section, you select the two points between which the spring or damper is connected.

Source Point

Select a **Source**. The default is **Select a connection point**. You will then do the actual selection in the **Source Point** subnode. If you select more than a single point, the spring will be connected to a virtual point, the location of which is the average of the selected points. If you make another choice than **Select a connection point**, the corresponding **Source Point** subnode will be removed.

The other option that is always available as **Source** is **Fixed**. When **Fixed** is selected, the source side of the spring is fixed in space, and not connected to any modeled parts. In this case, you need to specify the location of the source point. For **Connection point**, select **User defined** and enter the coordinate $\mathbf{X}_s$ of the source point.

If a **Base Motion** is defined in a Multibody Dynamics interface, it can be selected as **Source**. Selecting a base motion gives a behavior similar to using **Fixed**, with the exception that the base now can have a nonzero prescribed displacement, velocity, or acceleration.

Any **Attachment** node in the model can be selected as **Source**, irrespective of in which physics interface it is defined. The use of attachments required the Multibody Dynamics Module or the Rotordynamics Module.

From the **Source** list, you can also select any object that is of a rigid body type, irrespective on in which physics interface it is defined. Such objects are:

- **Rigid Material**
- **Spur Gear** (in the Multibody Dynamics interface)
- **Helical Gear** (in the Multibody Dynamics interface)
- **Bevel Gear** (in the Multibody Dynamics interface)
- **Worm Gear** (in the Multibody Dynamics interface)
- **Spur Rack** (in the Multibody Dynamics interface)
- **Helical Rack** (in the Multibody Dynamics interface)

Destination Point

Most settings are the same as for the **Source**. The differences are the following:

- The options Fixed and Base Motion are not available for the destination point.
- There is one other option: **Prescribed displacement**.

The available inputs for **Prescribed displacement** depend on the selected **Spring type** in the **Spring–Damper** section.

If the spring type is **Directional**, enter an expression for the **Prescribed displacement**, $\mathbf{u}_d$, and for the **Reference destination point**, $\mathbf{X}_d$. The values determine how the spring extension is computed.

If the spring type is **Matrix**, enter expressions for the **Prescribed displacement**, $\mathbf{u}_d$, and the **Prescribed rotation**, Θ_d.

Using **Prescribed displacement**, it is possible to connect the spring to another physics interface. In order to do that, you can enter expressions using **Nonlocal Couplings** defined in the **Definitions** node for a component.

When the **Spring type** is set to **Matrix**, the rotation of the source and the destination (Θ_s and Θ_d) are determined by availability of rotational degrees of freedom. If the physics interface itself has rotational degrees of freedom, these will be used in a point selection. For all physics interfaces, irrespective whether they have rotational degrees of freedom or not, a connection to an attachment or rigid body type object will provide a rotation.

Some settings in the GUI are only available if rotational degrees of freedom are present.

When the **Spring–Damper** node is used in the Multibody Dynamics interface, there is one more option: **Use selection filter**.

In large models, the list of available attachments can become very long. You can then create geometric filters to narrow down the search. When the **Use selection filter** checkbox is selected, two subnodes named Source Filter and Destination Filter are added to the **Spring–Damper** node. In these subnodes you can make graphic selections of the objects whose attachments should be shown in the **Source** and a **Destination** lists.

SPRING–DAMPER

Select a **Spring type** — **Directional** or **Matrix**. The latter option is not available for 1D and 1D axisymmetry.

Directional

Select a **Definition** — **Spring constant** or **Force as function of extension**.

- For **Spring constant** enter a value for k.
- For **Force as function of extension** enter an expression for F_s. The expression must be a function of the extension of the spring. The built-in variable for the spring extension has the form `<physicsName>.<SpringNodeTag>.dl`, for example `solid.spd1.dl`.

To add viscous damping in a dynamic analysis, enter a value or an expression for the **Damping coefficient** c.

Matrix

- Under **Spring constants**, enter the stiffness matrices defining the elastic connection between the source and destination. Input fields for the matrices $\mathbf{k}_u$ and $\mathbf{k}_\Theta$ are

always shown. Select **Translational-rotational coupling** to show input fields for the coupling matrices $\mathbf{k}_{u\Theta}$ and $\mathbf{k}_{\Theta u}$. In 2D, most elements of these matrices are by definition zero. Only elements that can be nonzero are shown; these are elements 13 and 23 for $\mathbf{k}_{u\Theta}$ and elements 31 and 32 for $\mathbf{k}_{\Theta u}$.

- Under **Damping coefficients**, enter the damping matrices defining the viscous connection between the source and destination. Input fields for the matrices $\mathbf{c}_u$ and $\mathbf{c}_\Theta$ are always shown. Select **Translational-rotational coupling** to show input fields for the coupling matrices $\mathbf{c}_{u\Theta}$ and $\mathbf{c}_{\Theta u}$. In 2D, most elements of these matrices are by definition zero. Only elements that can be nonzero are shown; these are elements 13 and 23 for $\mathbf{c}_{u\Theta}$ and elements 31 and 32 for $\mathbf{c}_{\Theta u}$.

By default, the relative displacement between source and destination is computed as $\Delta\mathbf{u} = \mathbf{u}_d - \mathbf{u}_s$. In some cases, however, it can be more reasonable to also include displacements resulting from a rigid body type rotation. If **Include rotational contribution in displacement** is selected, an additional term is added to the expression for the relative displacement. It describes the additional displacement due to the rotation of the destination if the source and destination were connected by a rigid bar element.

The **Translational-rotational coupling** checkbox and the input fields for the matrices $\mathbf{k}_\Theta$, $\mathbf{k}_{u\Theta}$, $\mathbf{k}_{\Theta u}$, $\mathbf{c}_\Theta$, $\mathbf{c}_{u\Theta}$, and $\mathbf{c}_{\Theta u}$ are not available in the Plate interface.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**. The latter option is not available for 1D or 1D axisymmetry.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions in the table for η_u based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

FREE LENGTH

This section is only available if **Directional** is selected as **Spring type**.

The free length is the distance between the connection points when there is no force in the spring. Select an option from the list: **Specify initial extension** or **Specify free length**.

- For **Specify initial extension** enter a value for Δl_0. The free length is computed as $l_f = l_0 - \Delta l_0$, where l_0 is the initial distance between the connection points.
- For **Specify free length** enter a value for l_f.

ACTIVATION CONDITIONS

If **Directional** is selected as **Spring type**, you can select a **Spring action** — **Bidirectional**, **Tension only**, or **Compression only**.

If you want to activate or deactivate the entire spring–damper, select the **Deactivation** checkbox. Then, enter a **Deactivation expression** i_{dac}. The expression is treated as a Boolean expression, so that when it evaluated to a nonzero value, the spring or damper is deactivated.

Select the **Permanently deactivate** checkbox if the spring is supposed to be removed permanently from the simulation when the deactivation condition is fulfilled for the first time.

As an example, if the spring should break at a certain extension, you can write an expression like `solid.spd2.dl>0.12[m]`, and select the **Permanently deactivate** checkbox. If the checkbox is not selected, the spring would become active again when its connection points come close enough to each other.

Spring–Damper Theory

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Spring–Damper**
Shell > **Spring–Damper**
Plate > **Spring-Damper**
Layered Shell > **Spring–Damper**
Membrane > **Spring–Damper**
Beam > **Spring–Damper**
Truss > **Spring–Damper**
Multibody Dynamics > **Spring–Damper**

Pipe Mechanics > **Spring–Damper**

Ribbon

Physics tab with **Solid Mechanics** selected:

Global > **Solid Mechanics** > **Spring–Damper**

Physics tab with **Shell** selected:

Global > **Shell** > **Spring–Damper**

Physics tab with **Plate** selected:

Global > **Plate** > **Spring–Damper**

Physics tab with **Layered Shell** selected:

Global > **Layered Shell** > > **Spring–Damper**

Physics tab with **Membrane** selected:

Global > **Membrane** > **Spring–Damper**

Physics tab with **Beam** selected:

Global > **Beam** > **Spring–Damper**

Physics tab with **Truss** selected:

Global > **Truss** > **Spring–Damper**

Physics tab with **Multibody Dynamics** selected:

Global > **Multibody Dynamics** > **Spring–Damper**

Physics tab with **Pipe Mechanics** selected:

Global > **Pipe Mechanics** > **Spring–Damper**

Source Point (for Spring–Damper)

Select a point, or a set of points, representing the source connection point for the Spring–Damper. The main intent is that you should select a single connection point, but it is also possible to select several points. If you select several points, the spring attachment location is the average value of the coordinates of the selected points. The displacement is computed as the average of the displacements in the selected points, which for a flexible body might not be the same as the actual displacement in the averaged location. The spring force is distributed over the selected points.

LOCATION IN USER INTERFACE

When **Select a connection point** is chosen under **Source** for in the parent Spring–Damper node, this subnode will automatically be generated. It cannot be added or removed manually.

Destination Point (for Spring–Damper)

Select a point, or a set of points, representing the destination connection point for the Spring–Damper. For details, see Source Point (for Spring–Damper).

LOCATION IN USER INTERFACE

When **Select a connection point** is chosen under **Destination** for in the parent Spring–Damper node, this subnode will automatically be generated. It cannot be added or removed manually.

Source Point Nodes

SOURCE POINT: BOUNDARY

Select a set of boundaries whose centroid represents the source connection point.

SOURCE POINT: EDGE

Select a set of edges whose centroid represents the source connection point.

SOURCE POINT: POINT

Select a set of points whose centroid represents the source connection point.

LOCATION IN USER INTERFACE

When **Centroid of selected entities** is chosen under **Connection point** for the source attachment in the parent Spring–Damper, a subnode will be generated based on the setting of **Entity level**. They cannot be added or removed manually.

Destination Point Nodes

DESTINATION POINT: BOUNDARY

Select a set of boundaries whose centroid represents the destination connection point.

DESTINATION POINT: EDGE

Select a set of edges whose centroid represents the destination connection point.

DESTINATION POINT: POINT

Select a set of points whose centroid represents the destination connection point.

LOCATION IN USER INTERFACE

When **Centroid of selected entities** is chosen under **Connection point** for the destination attachment in the parent Spring–Damper, a subnode will be generated based on the setting of **Entity level**. They cannot be added or removed manually.

Point Mass

Use the **Point Mass** node to add a discrete mass that is concentrated at a point.

POINT MASS

Enter a **Point mass** m.

The point mass matrix can be entered as **Isotropic**, **Diagonal**, or **Symmetric**. For **Isotropic**, the same mass is used in all the diagonal elements of the mass matrix. For all options except **Isotropic**, the orientations are defined by the selected coordinate system.

FRAME ACCELERATION FORCES

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Points** > **Point Mass**

Solid Mechanics, Explicit Dynamics> **Points** > **Point Mass**

Multibody Dynamics > **Points** > **Point Mass**

Ribbon

Physics tab with **Solid Mechanics** selected:

Points > **Solid Mechanics** > **Point Mass**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Points > **Solid Mechanics, Explicit Dynamics** > **Point Mass**

Physics tab with **Multibody Dynamics** selected:

Points > **Multibody Dynamics** > **Point Mass**

Added Mass

The **Added Mass** node is available on different geometrical entity levels and can be used to supply inertia that is not part of the material itself. Such inertia does not need to be isotropic in the sense that the inertial effects are not the same in all directions.

The **Added Mass** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Added Mass node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Added Mass, Interface node.

COORDINATE SYSTEM SELECTION

The added mass values are given with respect to the selected coordinate directions.

For more information about the special coordinate systems available in the Shell and Plate interfaces, see Added Mass in the documentation for these interfaces.

FACE DEFINING THE LOCAL ORIENTATIONS

This section is only present in the Added Mass node in the Shell interface.

ADDED MASS

Select a **Mass type** using Table 4-12 as a guide. The default option is the type for the geometric entity. The added mass matrix can be entered as **Isotropic**, **Diagonal**, or **Symmetric**. For **Isotropic** the same mass is used in all the diagonal elements of the mass matrix.

TABLE 4-12: AVAILABLE MASS TYPES BASED ON GEOMETRIC ENTITY.

MASS TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Mass per reference volume	ρ_V	kg/m^3	domains, boundaries
Mass per reference area	ρ_A	kg/m^2	domains, boundaries, edges
Mass per reference length	ρ_L	kg/m	edges
Total mass	m	kg	domains, boundaries, edges, points

FRAME ACCELERATION FORCES

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Mass, Spring, and Damper** > **Added Mass** (Domain, Boundary)
Solid Mechanics > **Edges** > **Added Mass**
Solid Mechanics > **Points** > **Added Mass**
Solid Mechanics, Explicit Dynamics > **Mass, Spring, and Damper** > **Added Mass** (Domain, Boundary)
Solid Mechanics, Explicit Dynamics > **Edges** > **Added Mass**
Solid Mechanics, Explicit Dynamics > **Points** > **Added Mass**
Shell > **Mass, Spring, and Damper** > **Added Mass** (Boundary, Edge)
Plate > **Mass, Spring, and Damper** > **Added Mass** (Domain, Boundary)
Layered Shell > **Mass, Spring, and Damper** > **Added Mass** (Boundary, Edge)
Layered Shell > **Points** > **Added Mass** (Point)
Layered Shell > **Mass, Spring, and Damper** > **Interfaces** > **Added Mass, Interface** (Boundary, Edge)

Membrane > **Mass, Spring, and Damper** > **Added Mass** (Boundary, Edge)
Beam > **Mass, Spring, and Damper** > **Added Mass** (Edge)
Truss > **Mass, Spring, and Damper** > **Added Mass** (Edge)
Truss, Explicit Dynamics > **Mass, Spring, and Damper** > **Added Mass** (Edge)
Wire > **Mass, Spring, and Damper** > **Added Mass** (Edge)
Multibody Dynamics > **Mass, Spring, and Damper** > **Added Mass** (Domain, Boundary)
Multibody Dynamics > **Edges** > **Added Mass**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **Mass, Spring, and Damper** > **Added Mass**
Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Solid Mechanics** > **Added Mass**

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Mass, Spring, and Damper** > **Added Mass**
Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Solid Mechanics, Explicit Dynamics** > **Added Mass**

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Mass, Spring, and Damper** > **Added Mass**

Physics tab with **Plate** selected:

Domains > **Mass, Spring, and Damper** > **Added Mass**
Boundaries > **Mass, Spring, and Damper** > **Added Mass**

Physics tab with **Layered Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Boundaries > **Interfaces** > **Added Mass, Interface**
Edges > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Interfaces** > **Added Mass, Interface**
Points > **Layered Shell** > **Added Mass**

Physics tab with **Beam**; **Truss**; **Truss, Explicit Dynamics**; or **Wire** selected:

Edges > **Mass, Spring, and Damper** > **Added Mass**

Physics tab with **Multibody Dynamics** selected:

Domains > **Mass, Spring, and Damper** > **Added Mass**

Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Multibody Dynamics** > **Added Mass**

Periodic Condition

Use a **Periodic Condition** to prescribe that the displacements on two different sets of boundaries with the same geometrical shape are related, as in a periodic structure.

Several different types of periodicity properties of the solution can be prescribed using this boundary condition.

- The **Continuity**, **Antiperiodicity**, and **User defined** periodic conditions directly prescribe relations between displacements and can be used for any type of study.
- The **Floquet periodicity** can be used for frequency domain problems with a spatial periodicity of the geometry and solution. The structure is typically a unit cell of a repetitive structure. It can also be used for linear buckling analysis of such structures.
- The **Cyclic symmetry** is a special case of a Floquet condition, intended for structures that consist of a number of sectors that are identical when rotated around a common axis, like a blade in a turbine. It can be used in frequency domain studies, as well as for linear buckling analysis.

The **Floquet periodicity** and **Cyclic symmetry** options are available only with some COMSOL products (see www.comsol.com/products/specifications/).

The two sets of boundaries between which there is a periodicity condition are called the *source* and *destination*. It is not required to have the same mesh on the source and destination, but the local accuracy of the solution at the boundaries will be better if you use the same mesh.

If you use physics-controlled meshing, the mesh on the source and destination boundaries will automatically match.

BOUNDARY SELECTION

The software usually automatically identifies the boundaries as either source boundaries or destination boundaries, as indicated in the selection list. This works fine for cases like opposing parallel boundaries. In other cases, right-click **Periodic Condition** and select **Manual Destination Selection** to control the destination. By default it contains the selection that COMSOL Multiphysics identifies.

In cases where the periodic boundary is split into several boundaries within the geometry, it might be necessary to apply separate periodic conditions to each pair of geometry boundaries for the matching to work properly.

DESTINATION SELECTION

This section is available for specifying the destination boundaries, if needed, when the **Manual Destination Selection** option is selected in the context menu for the **Periodic Condition** node. You can only select destination boundaries from the union of all source and destination boundaries.

PERIODICITY SETTINGS

With **Type of periodicity** you select the form of periodicity that your solution should have.

- For **Continuity** the displacements on the destination are set equal to the displacements on the source; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = \mathbf{u}(\mathbf{x}_\mathrm{s})$. If the source and destination boundaries are rotated with respect to each other, a transformation is automatically performed, so that corresponding displacement components are connected.
- For **Antiperiodicity** the displacements on the destination are set equal to the displacements on the source with the sign reversed; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = -\mathbf{u}(\mathbf{x}_\mathrm{s})$. If the source and destination boundaries are rotated with respect to each other, a transformation is automatically performed, so that corresponding displacement components are connected.
- For **Floquet periodicity** enter a **k-vector for Floquet periodicity** $\mathbf{k}_\mathrm{F}$. This is the wave number vector for the excitation.
- For **Cyclic symmetry** choose how to define the sector angle that the geometry represents using **Sector angle**. If **Automatic** is selected, the program attempts to find out how many full repetitions of the geometry there will be on a full revolution. If **User defined** is selected, enter a value for the sector angle θ_S. In both cases, also enter

an **Azimuthal mode number** for the mode to be studied. It can vary from 0 to $N/2$, where N is the total number of sectors on a full revolution.

- For **User defined** select the checkbox for any of the displacement components as needed. Then for each selection, choose the **Type of periodicity** — **Continuity** or **Antiperiodicity**. Each selected displacement component will be connected by $u_i(\mathbf{x}_d) = u_i(\mathbf{x}_s)$ or $u_i(\mathbf{x}_d) = -u_i(\mathbf{x}_s)$. If the source and destination boundaries are rotated with respect to each other, a transformation is automatically performed, so that corresponding displacement components are connected.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

EXCLUDED EDGES, EXCLUDED POINTS

To display these sections (**Excluded Edges** in 3D geometries only), click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For more information about these sections, see Excluded Surfaces, Excluded Edges, and Excluded Points in the *COMSOL Multiphysics Reference Manual*.

ORIENTATION OF SOURCE

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For information about the **Orientation of Source** section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

ORIENTATION OF DESTINATION

This section appears if the setting for **Transform to intermediate map** in the **Orientation of Source** section is changed from the default value, **Automatic**, and **Advanced Physics Options** is selected in the **Show More Options** dialog. For information about the **Orientation of Destination** section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

MAPPING BETWEEN SOURCE AND DESTINATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For information about the **Mapping Between Source and Destination** section, see Mapping Between Source and Destination in the *COMSOL Multiphysics Reference Manual*.

See also Cyclic Symmetry and Floquet Periodic Conditions.

Vibrations of an Impeller: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/impeller**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Connections** > **Periodic Condition**

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Connections** > **Periodic Condition**

Adiabatic Heating

Add an **Adiabatic Heating** node to model abrupt changes in temperature due to fast deformation. The increase in temperature is then defined by solving the distributed equation

$$\rho C_p \dot{T} = \beta_{ah} Q_h$$

here, ρ is the density, C_p is the heat capacity at constant pressure, T is the temperature field, β_{ah} is the coefficient of adiabatic heating, and Q_h corresponds to the heat sources due to mechanical dissipative processes.

The **Adiabatic Heating** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

INITIAL VALUES

Enter the **Initial temperature** T_{ini}. The default value is 293.15 K.

THERMODYNAMICS

The density ρ is taken from the material model (Linear Elastic Material, Nonlinear Elastic Material, or Hyperelastic Material).

The default **Heat capacity at constant pressure** C_p uses values **From material**. For **User defined**, enter an expression or value. The default value for the **User defined** is `0` J/(kg·K).

Enter the **Coefficient of adiabatic heating**, β_{ah}. The default value is 1 (dimensionless), which means that dissipative processes contribute 100% as heat sources.

Select the **Dissipative heat source** — **Include all dissipative sources** or **User defined**.

The **Dissipative heat source** list makes it possible to include specific heat sources for the adiabatic heating. Enter a value or expression for the heat source Q_h to include. For instance, the dissipated energy density due to creep is available under the variable `solid.Wc` and due to viscoplasticity under the variable `solid.Wvp`. Here `solid` denotes the name of the physics interface node.

TIME STEPPING

Select a **Method** — **Automatic**, **Backward Euler**, or **Domain ODEs**.

- The **Backward Euler** method is not available with the Layered Shell interface nor with the Layered Linear Elastic Material in the Shell and Membrane interfaces.

Automatic

The **Automatic** method corresponds to the backward Euler method except for the Layered Shell interface or when the Layered Linear Elastic Material is used. Domain ODEs are solved in these cases.

Backward Euler

For the **Backward Euler** method, enter the following settings:

- **Maximum number of local iterations**. To determine the maximum number of iteration in the Newton loop when solving the local equation.
- **Absolute tolerance**. To check the convergence of the local equation based on the step size in the Newton loop.
- **Relative tolerance**. To check the convergence of the local equation based on the step size in the Newton loop. The final tolerance is computed based on the current solution of the local variable and the entered value.
- **Residual tolerance**. To check the convergence of the local equation based on the residual.

If both a step size and residual convergence check is requested, it is sufficient that one of the conditions is fulfilled. Setting either the **Absolute tolerance** and **Relative tolerance** or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all are set to zero.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of this field can affect the convergence of the overall solution.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Adiabatic Heating**
Solid Mechanics, Explicit Dynamics > **More** > **Adiabatic Heating**
Membrane > **More** > **Adiabatic Heating**
Truss > **More** > **Adiabatic Heating**
Truss, Explicit Dynamics > **More** > **Adiabatic Heating**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics**selected:

Domains > **More** > **Adiabatic Heating**

Physics tab with **Membrane** selected:

Boundary > **More** > **Adiabatic Heating**

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **More** > **Adiabatic Heating**

Cell Periodicity

Use the **Cell Periodicity** node to model a periodic unit cell or RVE (*representative volume element*) of a larger repetitive structure. Periodic or homogeneous boundary conditions are then used on the cell outer boundaries.

If more than one **Cell Periodicity** node is used, they must have disjoint selections. It is thus possible to model more than one cell in the same study step.

From the **Cell Periodicity** node, it is possible to compute the elasticity matrix, the coefficients of thermal expansion or hygroscopic swelling for the equivalent homogenized material.

The **Cell Periodicity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). **Cell Periodicity** is available for 3D and 2D.

PERIODICITY SETTINGS

Select the **Formulation** — **Small strains** or **Large strains**.

Select the **Type** — **Periodic** or **Homogeneous**.

- When **Periodic** is selected, the average stresses or strains are periodic over the cell boundaries.
- The **Homogeneous** option can be used to prescribe boundaries when the cell geometry does not represent a periodic microstructure.

When **Small strains** is selected in the **Formulation** list, select the **Boundary conditions** for the unit cell — **Free expansion**, **Average strain**, **Average stress**, or **Mixed**. The **Average deformation gradient** is the only option when **Large strains** is selected.

Select the **Cell volume** — **Automatic**, **Void volume fraction**, or **User defined**.

- When **Automatic** is selected, the cell volume is computed from the domain or domains selected in the **Domain Selection** section.
- The **Void volume fraction** option can be used to scale the cell volume when there are voids that are not selected as domains. Enter the void volume fraction, f. This is the fraction of the cell that is occupied by voids. The volume of the cell is computed as

$$V = \frac{V_{\text{solid}}}{1-f}$$

- The **User defined** option allows to enter the cell volume V explicitly.

The variable `<item>.vol` contains the volume of the RVE used when computing average strains and stresses.

EFFECTIVE PROPERTIES

Depending on the choice of the **Boundary conditions** on the cell, different options are available.

- When **Free expansion** is selected from the **Boundary conditions** list, the cell is allowed to expand freely. This option is useful to determine the coefficients of thermal expansion or hygroscopic swelling in heterogeneous media.

  Select one or several of the available checkboxes — **Compute density**, **Compute coefficient of thermal expansion**, or **Compute coefficient of hygroscopic swelling**.

In 2D, use either a **Plane stress** or a **Generalized plane strain** approximation to calculate averaged properties with the **Free expansion** boundary condition.

- When **Average strain** is selected from the **Boundary conditions** list, it possible to derive homogenized elastic properties of perforated plates, porous media or composite structures. In this case, six load cases need to be examined in 3D.

  Enter values or expressions for the components of the **Average strain** tensor ε_{avg}. In a geometrically nonlinear analysis, the average strains are interpreted as Green–Lagrange strains.

  Select one or several of the available checkboxes — **Compute density**, **Compute elasticity matrix, standard notation**, or **Compute elasticity matrix, Voigt notation**.

- When **Average stress** is selected from the **Boundary conditions** list, it is possible to derive the homogenized compliance matrix. Enter values or expressions for the components of the **Average stress** tensor σ_{avg}. In a geometrically nonlinear analysis, the stresses are interpreted as second Piola–Kirchhoff stresses.

  Select one or several of the available checkboxes — **Compute density**, **Compute elasticity matrix, standard notation**, or **Compute elasticity matrix, Voigt notation**.

In 2D, use a **Generalized plane strain** approximation to calculate averaged properties with the **Average strain** or **Average stress** boundary conditions.

- When **Mixed** is selected from the **Boundary conditions** list, enter either the **Average strain** or **Average stress** for each tensor component.

	When using a **Plane stress** or a **Plane strain** 2D approximation with the **Mixed** boundary condition, you only need to enter three components (XX, YY, XY) of the average strain or average stress tensors. For the **Generalized plane strain** approximation, you also need to enter the ZZ component.

Load Group, Material, and Study Generation

You can create the elasticity matrix for a homogenized material, and make it accessible as a material to be used in other components. In order to do this, a number of fundamental load cases must be analyzed. The necessary definitions and steps required for such an analysis can be automated as described below.

On the **Periodicity Settings** section toolbar, there is an icon **Automated Model Setup** ( ). It has a list with three entries:

- **Create Load Groups and Study** . This option can be selected when the **Boundary conditions** are **Average strain** or **Average stress**. When you select it, the following changes will be made to the model:
  - A number of load groups will be created under **Global Definitions**. They are collected in a group named **Load Groups for Cell Periodicity**. The load groups correspond to unit loads along different axes.
  - The **Average strain** or **Average stress** tensor in the **Cell Properties** section will be populated using the load group variables.
  - A new study, named **Cell Periodicity Study**, will be created. This study contains a stationary study step. In the **Study Extensions** section of the new **Cell Periodicity Study**, one load case is added for each load group.
- **Create Material by Reference** ( ). This option can be selected when the **Boundary conditions** are **Average strain** or **Average stress**. When you select it, a new material will be created under **Global** > **Materials**. It contains the elasticity matrix or the compliance matrix. The name of this material is **Homogeneous Material**.
- **Create Material by Value** ( ). This option can be selected when the **Boundary conditions** are **Average strain** or **Average stress**. This option is similar to **Create Material by Reference**, with the difference that the material properties are numerical values, and not references to variables in the node. The benefits of this option is faster computation when the material is used by other physics interfaces. Also, if the **Cell**

Periodicity node is deleted, the homogenized material generated from it can still be used in subsequent computations.

ADVANCED

To enable this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. Furthermore, this section is displayed only when either **Average strain** or **Average stress** is selected from the **Boundary conditions** list.

If elasticity or compliance matrices are requested in the **Effective Properties** section, then an input for **Solution tag** is displayed. You can modify the behavior of how the studies for computing the homogenized matrices are generated by selecting a **Solution tag** — **Automatic** or **Legacy**. The **Legacy** option is mainly for maintaining compatibility with versions prior to 6.0. With the default **Automatic** option, it is, for example, possible to set up nested homogenizations. A unique tag for the study will be composed by combining the physics interface tag and the tag of the **Cell Periodicity** node, for example `solid1cp1std`.

Specify if the cell periodicity study should contain a parametric sweep. Select **Add parametric sweep** — **Yes** or **No**.

If **Yes** is selected, specify the **Sweep type** — **Specified Combinations** or **All combinations**.

For **Specified Combinations**, select the **Parameter selection for material** — **First**, **Last**, or **Manual**. By default, the last parameter solved for dictates the homogenized material properties. Enter the **Parameter index** when **Manual** is selected from the list.

Use the **Parameters** table to add parameters, their ranges, and their units for the parametric solver. When **All combinations** is selected in the **Sweep type** list, and **Manual** is selected in the **Parameter selection for material** list, enter the parameter index and the total number of parameters to solve for. The settings are similar to the Parametric Sweep study step.

!	For this type of analysis to work correctly, it is important that you do not edit the generated nodes manually. By clicking the **Create Load Groups and Study** button again, you can reset all settings in the generated nodes to their default values.

See also

- Load Cases and Effective Properties of Periodic Structures in the *Structural Mechanics Modeling* chapter.
- Cell Periodicity in the *Structural Mechanics Theory* chapter.
- Materials and Solid Mechanics Material Properties in the *COMSOL Multiphysics Reference Manual*.

- *Micromechanical Model of a Fiber Composite*: Application Library path **Structural_Mechanics_Module/Material_Models/micromechanical_model_of_a_fiber_composite**
- *Micromechanics and Macromechanics of a Composite Cylinder*: Application Library path **Composite_Materials_Module/Tutorials/composite_cylinder_micromechanics_and_stress_analysis**

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Cell Periodicity**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **More** > **Cell Periodicity**

Boundary Pair

In the **Boundary Pair** node, you supply information about the corresponding periodic boundaries. In 2D, you need at least two such pairs, and in 3D at least three pairs. The number of pairs depends on the shape of the unit cell. This node has no other settings that the boundary selection.

Select all corresponding boundaries belonging to both sides of the pair. If you need to manually specify the source and destination sides of the pairs, add a **Destination Selection** subnode.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Cell Periodicity** > **Boundary Pair**

Ribbon

Physics tab with **Cell Periodicity** node selected in the model tree:

Attributes > **Boundary Pair**

Low-Reflecting Boundary

Use the **Low-Reflecting Boundary** node to let waves pass out from the model without reflection in time-dependent or frequency domain analysis. As a default, it takes material data from the domain in an attempt to create a perfect impedance match for both pressure waves and shear waves. It can be sensitive to the direction of the incoming wave.

The **Low-Reflecting Boundary** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). **Low-Reflecting Boundary** is available for 3D, 2D, and 2D axisymmetry.

DAMPING

Select a **Damping type** — **P and S waves** or **User defined**. For **User defined** enter values or expressions for the **Mechanical impedance** $\mathbf{d}_{\mathrm{im}}$. The defaults for all values are `0.5*solid.rho*(solid.cp+solid.cs)`.

Low-Reflecting Boundary Condition

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Low-Reflecting Boundary** (Boundary)
Solid Mechanics, Explicit Dynamics > **More** > **Low-Reflecting Boundary (Boundary)**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More** > **Low-Reflecting Boundary**

Fasteners

The **Fasteners** node is designed to enable simplified modeling of fasteners such as rivets and bolts. It assumes that the corresponding fastener holes are already defined in the geometry. Typically, two plates are connected using an elastic connector. Two types of such connections are available:

- The edges of matching holes are connected.
- Boundaries surrounding the matching holes are connected. Such boundaries should typically match the size of the rivet heads or washers.

The latter type of connection offers greater accuracy and helps reduce local stress concentrations, especially when bolts are modeled. Creating such connections requires a more detailed geometric representation, in which the fastener holes are enclosed by appropriate, typically annular, boundaries.

The actual selection of the pairs of holes to be connected is done in the Hole Selection subnode.

When one or more **Fasteners** nodes are active in the model, five entries will be added to the list of **Result Templates**:

- A **Fasteners Forces** plot group that shows the normal and shear forces at each fastener and a label for the identity number of each fastener. The fasteners forces have to be evaluated first in the **Fasteners Normal Forces** and the **Fasteners Shear Forces** evaluation groups. The coloring can be affected by the addition of a **Safety** subnode.
- A **Fasteners Normal Forces** evaluation group that contains a table of the location, the normal forces, and the damage index of all fasteners. Evaluate this table to display the forces in the **Fasteners Forces** plot group.
- A **Fasteners Shear Forces** evaluation group that contains a table of the location, the shear forces, and the damage index of all fasteners. Evaluate this table to display the forces in the **Fasteners Forces** plot group.

- A **Fasteners Forces Magnitude** evaluation group that contains a table of the normal and shear forces magnitude of all fasteners.
- A **Connected Region Indicator** plot group that contains a line plot or a surface plot showing the hole edges or hole surrounding boundaries, respectively, on the destination surface where the fasteners are active.

You must evaluate the **Fasteners Normal Forces** and **Fasteners Shear Forces** evaluation groups to update the **Fasteners Forces** plot group if you change the number of fasteners in your model and compute new results.

Also, you need to remove the **Connected Region Indicator** plot group and replace it by a new one if you change the hole selection type between edge or boundary.

FASTENER PROPERTIES

Enter

- **Diameter**, d_f for the fasteners diameter.
- **Plate thickness, destination** t_{dst} for the plate thickness on the destination side.

From the **Plate thickness, source** list select **Same as destination** or **User defined** to reuse the thickness as t_{dst} or to enter a different value for t_{src}.

The stiffness of the fasteners can be computed in two ways:

- Use analytical expressions when the fastener can be approximated as a beam.
- Enter the shear and normal stiffness values.

From the **Fastener model** list, select **Beam** or **Spring**.

Beam

The normal stiffness, k_{n}, and shear stiffness, k_{s}, of the fastener are computed using the following expressions:

$$k_{\mathrm{n}} = \frac{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}{4(t_{\mathrm{dst}} + t_{\mathrm{src}})}$$

$$k_{\mathrm{s}} = \frac{1}{C_{\mathrm{b}} + C_{\mathrm{s}} + C_{\mathrm{br}} + C_{\mathrm{pbr}}}$$

The four compliance contributions are defined as

$$C_{\mathrm{b}} = \alpha_{\mathrm{b}} \frac{3t_{\mathrm{src}}^3 + 16t_{\mathrm{src}}^2 d_{\mathrm{dst}} + 16t_{\mathrm{dst}}^2 d_{\mathrm{src}} + 3t_{\mathrm{dst}}^3}{2\pi E_{\mathrm{f}} d_{\mathrm{f}}^4}$$

$$C_{\mathrm{s}} = \alpha_{\mathrm{s}} \frac{7 + 14\nu_{\mathrm{f}} + 8\nu_{\mathrm{f}}^2}{2(1+\nu_{\mathrm{f}})} \cdot \frac{t_{\mathrm{dst}} + t_{\mathrm{src}}}{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}$$

$$C_{\mathrm{br}} = \alpha_{\mathrm{br}} \left(\frac{1}{t_{\mathrm{src}} E_{\mathrm{f}}} + \frac{1}{t_{\mathrm{dst}} E_{\mathrm{f}}} \right)$$

$$C_{\mathrm{pbr}} = \alpha_{\mathrm{pbr}} \left(\frac{1}{t_{\mathrm{src}} E_{\mathrm{src}}} + \frac{1}{t_{\mathrm{dst}} E_{\mathrm{dst}}} \right)$$

In these expressions

- d_{f} is the diameter of the fastener.
- E_{f}, and ν_{f} are the elastic modulus and Poisson's ratio of the fastener, respectively.
- E_{src}, and E_{dst} are the elastic moduli of the shell on the source side and destination side, respectively.

- t_{src} and t_{dst} are the thickness of the plate on the source and destination sides, respectively.
- The multipliers α_m make it possible to adjust the default analytical values.

Enter the fasteners data:

- **Young's modulus**, E_{f}.
- **Poisson's ratio**, ν_{f}.
- **Bending flexibility multiplier**, α_{b}.
- **Shear flexibility multiplier**, α_{s}.
- **Fastener bearing flexibility multiplier**, α_{br}.
- **Plate bearing flexibility multiplier**, α_{pbr}.

User Defined

Enter the **Normal stiffness**, k_{n}, and **Tangential stiffness**, k_{s}.

Select **Include pretension** to add a load to the fastener normal force definition.

PRETENSION

This section is only visible when **Include pretension** is selected.

When pretension is included, the normal stiffness, k_{n}, is corrected to include an estimated head stiffness by the following expression:

$$k_{\mathrm{n}} = \frac{\pi E_{\mathrm{f}} d_{\mathrm{f}}^2}{4\left(t_{\mathrm{dst}} + t_{\mathrm{src}} + \frac{9}{10} d_{\mathrm{f}}\right)}$$

Enter the **Pretension force**, F_{pre}.

The **Grip stiffness** can be specified in two ways: either by using analytical expressions based on the simplified pressure-cone method or by directly entering stiffness values.

From the list, select **Shigley's frustum, 1 material**; **Shigley's frustum, 2 materials**; or **User defined**.

Select **Shigley's frustum, 1 material** when the plates have the same Young's modulus. Select **Shigley's frustum, 2 materials** when the Young's moduli are different.

When both plates have the same material, the grip stiffness is evaluated as

$$k_g = \frac{\pi E d_h \tan\alpha_{sh}}{2\log\left(\frac{(t_{\mathrm{dst}}+t_{\mathrm{src}})\tan\alpha_{sh}+d_w-d_h}{(t_{\mathrm{dst}}+t_{\mathrm{src}})\tan\alpha_{sh}+d_w+d_h}\cdot\frac{d_w+d_h}{d_w-d_h}\right)}$$

For **Shigley's frustum, 1 material** or **Shigley's frustum, 2 materials** enter

- **Washer diameter**, d_{w}.
- **Hole diameter**, d_{h}.
- **Frustum angle**, α_{sh}.

For **User defined** enter values or expressions for the **Grip stiffness**, k_{g}

- Simplified Modeling of Fasteners
- Theory for Fasteners

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Fasteners**

Ribbon

Physics tab with **Solid Mechanics** selected:

Global > Fasteners

Safety (for Fasteners)

Add a **Safety** subnode to a **Fasteners** node in order to evaluate the safety margin against failure of the individual fasteners.

SAFETY

Enter

- The **Critical normal force**, $F_{\mathrm{n,max}}$
- The **Normal force failure exponent**, α_{n}
- The **Critical shear force**, $F_{\mathrm{s,max}}$
- The **Shear force failure exponent**, α_{s}

The failure criterion is assumed to be a power law of the type

$$\left(\frac{F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} = 1$$

In the **Warning Level** section, select either **Safety factor** or **Margin of safety**. Then enter either the safety factor sf_0, or the margin of safety, ms_0. The relation between the two is

$$\mathrm{sf}_0 = 1 + \mathrm{ms}_0$$

When using any of these option, a warning level is defined by the criterion

$$\left(\frac{\mathrm{sf}_0 F_{\mathrm{n}}}{F_{\mathrm{n,\,max}}}\right)^{\alpha_n} + \left(\frac{\mathrm{sf}_0 F_{\mathrm{s}}}{F_{\mathrm{s,\,max}}}\right)^{\alpha_s} \geq 1$$

The purpose of the settings in this section is to make it possible to flag fasteners as dangerous, even if the load is less that the failure load.

- Simplified Modeling of Fasteners
- Theory for Fasteners

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Fasteners** > **Safety**

Shell > **Fasteners** > **Safety**

Ribbon

Physics tab with **Fasteners** node selected in the model tree:

Attributes > **Safety**

Hole Selection

The **Hole Selection** node is used to specify pairs of holes that will be connected by fasteners. The connection between the two shell boundaries can be defined either by referencing the hole edges directly or by using the boundaries surrounding the holes. Each setup includes at least one **Hole Selection** subnode, and additional subnodes can be added as needed.

By default, matching hole pairs are identified automatically, so a single **Hole Selection** node is usually sufficient. When using manual selection, however, you must add one **Hole Selection** node for each hole pair.

EDGE SELECTION, SOURCE

This section is enable when **Source selection type** is set to **Edge**.

The way the edge selections are used depends on whether automatic or manual edge selection is used.

When an **Edge Selection** node is added, the selection defaults to **All edges**, which implies that all edges on the source selection in the parent **Fasteners** node are shown as applicable. This is the standard configuration when working in automatic mode. You can, however, remove some edges from the list.

When working in manual mode, select a set of edges that define the hole on the source side.

In either case, all edges around a hole must be selected for the hole to be defined. The **Group by continuous tangent** option is a useful tool for making fast and correct manual selections of holes.

EDGE SELECTION, DESTINATION

This section is enable when **Destination selection type** is set to **Edge**.

The selection of destination side hole edges works the same way as described for Edge Selection, Source.

BOUNDARY SELECTION, SOURCE

This section is enable when **Source selection type** is set to **Boundary**.

The way the boundary selections are used depends on whether automatic or manual edge selection is used.

When a **Hole Selection** node is added, the selection defaults to **All boundaries**, which implies that all boundaries from the source selection in the parent **Fasteners** node are shown as applicable. This is the standard configuration when working in automatic mode. However, you can manually remove specific boundaries from the list if you do not want them to be included.

When working in manual mode, select a set of boundaries that surround the hole on the source side.

In either case, all boundaries around a hole must be selected in order for a hole to be defined.

BOUNDARY SELECTION, DESTINATION

This section is enable when **Destination selection type** is set to **Boundary**.

The selection of destination side hole boundaries works the same way as described for Boundary Selection, Source.

SELECTION TYPE

From the **Source selection type** list, select **Edge** or **Boundary**. This defines the type of connection that is made on the source shell boundaries.

From the **Destination selection type** list, select **Edge** or **Boundary**. This defines the type of connection that is made on the destination shell boundaries.

In Shell interface, **Edge** is set as default selection type for both source and destination selection.

FASTENER HOLE DETECTION

Select a method for detecting the fastener holes, **Automatic** or **Manual**. For **Automatic**, you can also adjust the **Diameter tolerance**, **Offset tolerance**, **Maximum angle**, and **Annular center offset**. These values control when two holes are deemed to form a pair. The last parameter is available only when the selection type for either source or destination is set to boundary and the checkbox **Search annular shape only** is selected. Selecting this checkbox adds the requirement that the region around a hole is actually annular. When not selected, any boundary around a hole is valid as the connecting boundary.

The exact definitions are given in Automatic Hole Detection in the theory chapter.

In automatic mode, the source and destination boundaries selected in the parent **Fasteners** node are scanned to detect all holes within the active selections. Hole pairs are then identified if the holes on both sides have diameters within the specified tolerance relative to the fastener diameter and if their centers are aligned along the same axis normal to the two boundaries.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Fasteners** > **Hole Selection**

Ribbon

Physics tab with **Fasteners** node selected in the model tree:

Attributes > **Hole Selection**

Rigid Connector

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. A rigid connector can connect an arbitrary combination of boundaries, edges, and points that all will move together as being attached to a virtual rigid object.

You can add the **Rigid Connector** node at the boundary, edge, and point levels.

- When added at the boundary level, you can connect boundaries, edges (3D), and points as long as at least one boundary is selected. Selecting edges and points is optional.
- When added at the edge level (3D only), you can connect edges and points as long as at least one edge is selected. Selecting points is optional.
- When added at the point level, you can connect a set of points to each other.

When the selection consists of boundaries only, you can also choose to remove the assumption of rigidity, while still respecting force and moment equilibrium. With this formulation, it is possible to avoid artificial constraint effects at the connected boundaries.

If the study step is geometrically nonlinear, the rigid connector takes finite rotations into account.

Rigid connectors are available in the Solid Mechanics, Multibody Dynamics, Shell, Beam, and Pipe Mechanics interfaces. Rigid connectors from different interfaces can be attached to each other.

You can add functionality to the rigid connector through the following subnodes:

- Applied Force (Rigid Connector) to apply a force in given point.
- Applied Moment (Rigid Connector) to apply a moment.
- Mass and Moment of Inertia (Rigid Connector) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Connector) to add a translational or rotational spring or damper in a given point.

When physics symbols are shown, a rigid connector is represented by a symbol at the center of rotation, together with a set of lines connecting the center of rotation with the centers of gravity of the selected entities.

- When the rigid connector is added at the boundary level, such symbols are shown only for the selected boundaries, but not for auxiliary selections of edges or points.
- When the rigid connector is added at the edge level, such symbols are shown only for the selected edges, but not for auxiliary selections of points.
- Because of the way physics symbols are evaluated, as a lightweight operation when moving between physics nodes in the model builder tree, it is sometimes not possible to determine the center of rotation. In particular, if an offset is supplied, it will not be taken into account.

The **Rigid Connector** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). **Rigid Connector** is available for 3D and 2D.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Rigid Connector node.

INTERFACE SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the Rigid Connector, Interface node.

BOUNDARY SELECTION

This section is present when the **Rigid Connector** node has been added at the boundary level. Select one or more boundaries to be part of the rigid region.

EDGE SELECTION

This section is present in 3D when the **Rigid Connector** node has been added at the boundary or edge level, and **Connection Type** is **Rigid**.

- When the **Rigid Connector** is added at the edge level, select one or more edges that form part of the rigid region.
- When the **Rigid Connector** is added at the boundary level, this section is initially collapsed. Here, you can add optional edges to the rigid region. The edges cannot be adjacent to the selected boundaries.

POINT SELECTION

This section is always present when **Connection Type** is **Rigid**.

- When the **Rigid Connector** is added at the point level, select a number of points that form the rigid region.
- When the **Rigid Connector** is added at the boundary or edge levels, this section is initially collapsed. Here, you can add optional points to the rigid region. The points cannot be adjacent to the selected boundaries or edges.

If your selection is too small, you can introduce rigid body motions in the model. No exact rules can be given because there are many possible configurations where, for example, domains are connected to each other or affected by other constraints. Also, you can, by providing proper constraints for the rigid connector, suppress rigid body motions.

Some suggestion for ensuring stability are

- At the point level in 3D, at least three points, not located on a straight line, are needed.
- At the point level in 2D, at least two points are needed.
- At the edge level in 3D, a single straight edge is not sufficient since it can act as a hinge.

A set of consistency checks where violations of these rules can be detected is performed. You can suppress these checks by clearing the **Include consistency checks** checkbox in the **Advanced** section.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed displacements and rotations are specified along the axes of this coordinate system. It is also used for defining the axis directions of the moment of inertia tensor of the **Mass and Moment of Inertia** subnode.

In order to be applicable, a coordinate system must have axis directions that are independent of the location in space.

CONNECTION TYPE

Select **Rigid** or **Flexible**. When the connection type is rigid, the whole rigid connector acts as a virtual rigid object. In the flexible formulation, the selected boundaries are allowed to have internal deformations, and the kinematic constraints are fulfilled only in an average sense. The flexible formulation is useful, for example, when applying loads because it will reduce local constraint effects.

If a selected boundary is located on a rigid domain, the connection type setting does not matter. The rigid formulation is always used,

The flexible formulation can be used when the selection only consists of boundaries.

- The **Connection Type** section is shown only when the **Rigid Connector** has been added at the boundary level.
- When the connection type has been set to **Flexible**, the **Edge Selection** and **Point Selection** sections are hidden.
- Any edges or points that have been selected prior to changing from **Rigid** to **Flexible** will be ignored.

With the flexible formulation, some extra degrees of freedom are added for each rigid connector. In order to get good convergence properties for a nonlinear solution, these variables must have a good scaling.

When a new solver sequence is generated, then an appropriate manual scaling is automatically set for these variables.

If, however, an existing solver sequence was generated while **Connection Type** was set to **Rigid**, and you then change to **Flexible**, no such scaling will be present. In this case, you either have to regenerate the solver sequence or set the scaling manually under the **Dependent Variables** node in each study step.

CENTER OF ROTATION

The center of rotation serves two purposes.

- If you prescribe the displacement of the rigid connector, this is the location in space where it is fixed.
- Results are interpreted with respect to the center of rotation.

Select a **Center of rotation** — **Automatic**, **Centroid of selected entities**, or **User defined**.

- For **Automatic**, the center of rotation is at the geometrical center of the selected geometrical objects of the highest geometrical dimension.

- For **Centroid of selected entities** select an **Entity level** — **Boundary**, **Edge**, or **Point**. The available choices depend on the physics interface and geometrical dimension.

- When selected, a default Center of Rotation: Boundary, Center of Rotation: Edge, or Center of Rotation: Point subnode is automatically added.
- The center of rotation is located at the centroid of the selected entities, which do not need to be related to the rigid connector itself. As a special case, you can select a single point, and use it as center of rotation.
- When **Entity level** is set to **Point**, any point in the geometry can be selected, even if it is not part of the physics interface.

- For **User defined**, in the **Global coordinates of center of rotation** X_C table enter coordinates based on space dimension.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{C}} = \mathbf{X}_{\text{C,input}} + \mathbf{X}_{\text{offset}}$$

PRESCRIBED DISPLACEMENT AT CENTER OF ROTATION

To define a prescribed displacement at the center of rotation for each space direction, select one or several of the available checkboxes then enter values or expressions for the prescribed displacements. The direction coordinate names can vary depending on the selected coordinate system.

- **Prescribed in x direction** $u_{0\text{x}}$
- **Prescribed in y direction** $u_{0\text{y}}$
- *For 3D components:* **Prescribed in z direction** $u_{0\text{z}}$

PRESCRIBED ROTATION

Specify the rotation at the center of rotation. Select from the **By** list: **Free**, **Constrained rotation**, or **Prescribed rotation** about the center of rotation.

For 2D components, the **Constrained rotation** and **Prescribed rotation** is always about the *z*-axis, so no component selection is necessary.

Constrained Rotation (3D Components)

For **Constrained rotation** select one or more of the available checkboxes to enforce zero rotation about the corresponding axis in the selected coordinate system:

- **Constrain rotation about x-axis**
- **Constrain rotation about y-axis**
- **Constrain rotation about z-axis**

Prescribed Rotation

For 3D components, enter an **Axis of rotation** vector, Ω.

Then, select **Prescribed motion through** — **Angle of rotation** or **Angular velocity**. For **Angle of rotation**, enter the angle ϕ_0. For **Angular velocity**, enter the angular velocity ω_0.

RELEASED DEGREES OF FREEDOM

In some cases it can be useful to not constrain the displacement in a certain direction. For instance, the radial displacement of a Rigid Connector acting on a circular cross-section could be allowed to be free. To do so, select a local **Coordinate system** for specifying the directions in which the degree of freedom will be released. The **Coordinate system** list contains only applicable coordinate systems in the model.

Select one or more of the available checkboxes to release the displacement in the corresponding axis in the selected coordinate system:

- **Release displacement in x1 direction**

- **Release displacement in x2 direction**
- *For 3D components:* **Release displacement in x3 direction**

!	Note that the **Rigid Connector** introduces its own global displacement degrees of freedom (DOFs) and global rotation DOFs. When certain displacement field components are released, this reduces the number of equations used to solve for the rigid connector's global DOFs, while preserving the number of said DOFs. Therefore, it may become necessary to constrain some of the Rigid Connector's global DOFs to achieve static determinacy.

The section **Released Degrees of Freedom** is only shown if **Connection Type** is **Rigid** (if applicable), and if the checkbox **Use weak constraints for rigid-flexible connection** in the section **Constrain Settings** is not enabled.

REACTION FORCE SETTINGS

Select **Evaluate reaction forces using weak constraints** to compute the reaction force caused by a prescribed motion. The default is to not compute the reaction force. When selected, the prescribed motion is implemented as a weak constraint.

Select **Apply reaction only on rigid body variables** to use a unidirectional constraint for enforcing a prescribed motion. The default is that bidirectional constraints are used. This setting is useful in a situation where a bidirectional constraint would give an unwanted coupling in the equations. This would happen if the prescribed value of the motion is a variable solved for in other equations.

FORMULATION

Some contributions from a rigid connector will, under geometric nonlinearity, result in a nonsymmetric local stiffness matrix. If all other aspects of the model are such that the global stiffness matrix would be symmetric, then such a nonsymmetric contribution may have a heavy impact on the total solution time and memory usage. In such cases, it is often more efficient to use an approximative local stiffness matrix that is symmetric.

Select **Use symmetric formulation for geometric nonlinearity** to force all matrix contributions from the rigid connector and its subnodes to be symmetric.

Using an approximative stiffness matrix will in general require more iterations. However, since the computational cost per iteration will be cut at least by a factor of two if a symmetric matrix can be used, it is usually more efficient to ignore a weak nonsymmetry.

In particular, if the rotation of the rigid connector per time step or parameter increment is small, there will in general be no increase in the number of iterations at all if this option is used.

When the global stiffness matrix is nonsymmetric for other reasons, then there is nothing to be gained from symmetrizing the contribution from the rigid connector.

CONSTRAINT SETTINGS

On the boundaries where the rigid connector is coupled to a flexible material, all nodes on such a boundary are constrained to move as a rigid body. As a default these constraints are implemented as pointwise constraints. If you want to use a weak constraint formulation, select **Use weak constraints for rigid-flexible connection**.

This formulation cannot be combined with the **Flexible** formulation of the rigid connector, which in itself is a special form of weak constraint.

ADVANCED

When the rigid connector is added at edge or point level, automatic tests are performed to check for selections that would result in a singularity. If these checks give false positives, you can turn them off by clearing the **Include consistency checks** checkbox. This could, for example, be necessary if a rigid connector, which in itself is singular, is connected to another one in a way that forms a stable configuration.

When the **Rigid Connector** is added at the point level, select the **Add rotational stiffness for two-point selection** checkbox to automatically suppress singularities when the **Include consistency checks** checkbox is disabled. This checkbox allows to enter a rotational stiffness, k_θ, to stabilize the rigid connector. The rotational stiffness is only added if less than three points are selected.

Select **Group dependent variables in solver** — **From physics interface** (default), **Yes**, or **No**, to choose how to group the dependent variables added by the **Rigid Connector** feature in the solver sequence.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can activate and deactivate the rigid connector by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.
- You can assign the value of the prescribed displacement and rotation to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

- Rigid Connector
- Harmonic Perturbation
- Load Cases

Assembly with a Hinge: Application Library path
Structural_Mechanics_Module/Connectors_and_Mechanisms/hinge_assembly

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Connections** > **Rigid Connector**
Solid Mechanics > **Edges** > **Rigid Connector**
Solid Mechanics > **Points** > **Rigid Connector**
Layered Shell > **Face and Domain Constraints, Interfaces** > **Rigid Connector, Interface**
Layered Shell > **Connections** > **Rigid Connector**
Multibody Dynamics > **More Constraints** > **Rigid Connector**
Multibody Dynamics > **Edges** > **Rigid Connector**
Multibody Dynamics > **Points** > **Rigid Connector**

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **Connections** > **Rigid Connector**

Edges > **Solid Mechanics** > **Rigid Connector**
Points > **Solid Mechanics** > **Rigid Connector**

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Rigid Connector, Interface**
Edges > **Connections** > **Rigid Connector**

Physics tab with **Multibody Dynamics** selected:

Boundaries > **More Constraints** > **Rigid Connector**
Edges > **Multibody Dynamics** > **Rigid Connector**
Points > **Multibody Dynamics** > **Rigid Connector**

Center of Rotation Nodes

For each of the following physics features, a subnode is automatically added as indicated.

- Rigid Connector
- Initial Values (Rigid Material)
- Prescribed Displacement/Rotation
- Average Rotation

CENTER OF ROTATION: DOMAIN

Use the **Center of Rotation: Domain** subnode to select a set of domains whose centroid represents the center of rotation.

CENTER OF ROTATION: BOUNDARY

Use the **Center of Rotation: Boundary** subnode to select a set of boundaries whose centroid represents the center of rotation.

CENTER OF ROTATION: EDGE

Use the **Center of Rotation: Edge** subnode to select a set of edges whose centroid represents the center of rotation.

CENTER OF ROTATION: POINT

Use the **Center of Rotation: Point** subnode to select a set of points whose centroid represents the center of rotation.

LOCATION IN USER INTERFACE

These nodes are automatically added when **Centroid of selected entities** is selected in the parent node. They cannot be added or removed manually.

Thermal Expansion (Rigid Connector)

Add the **Thermal Expansion** subnode to prescribe a deformation of the rigid connector caused by changes in temperature. This makes it possible to reduce stresses caused by the rigid connector being rigid, while there are thermal deformations in the flexible body to which it is attached.

Select an **Input type** to specify how the thermal strain is entered. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

here, the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

SHELL PROPERTIES

This section is only present in the Layered Shell interface, where it is described in the documentation for the Thermal Expansion (Rigid Connector) node.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain in the rigid connector.

Select **Inherit from domain** to take the thermal expansion data from the domain to which it is attached. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual material in the rigid connector has the same thermal expansion as the domain itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for the **Volume reference temperature** T_{ref} that is the temperature at which there are no thermal displacements at the constraints.
- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any temperature variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The default values **From material** are used. This requires that a material has been assigned to the boundaries, edges, or points where the constraint is active. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the *Structural Mechanics Modeling* chapter.
- Thermal Expansion of Constraints in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Connector** > **Thermal Expansion**

Layered Shell > **Rigid Connector** > **Thermal Expansion**
Layered Shell > **Rigid Connector, Interface** > **Thermal Expansion**
Multibody Dynamics > **Rigid Connector** > **Thermal Expansion**

Ribbon
Physics tab with **Rigid Connector** or **Rigid Connector, Interface** node selected in the model tree:

Attributes > **Thermal Expansion**

Applied Force (Rigid Connector)

Use the **Applied Force** to apply a force to the rigid connector. The force can act at an arbitrary position in space, and if it is not located at the center of rotation, there is also a moment contribution. The force that you specify is interpreted in the selected coordinate system.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model. Select a **Coordinate system** for specifying the directions of the force.

In order to be applicable, a coordinate system must have axis directions that are independent of the location in space.

LOCATION

Select an option from the list: **Center of rotation** or **User defined**. This is the location where the force is applied.

- For **Center of rotation**, the location of the load is taken as the center of rotation as defined in the parent **Rigid Connector** node.
- For **User defined**, enter the **Location (global coordinates)**, $\mathbf{X}_p$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_{offset}.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$

APPLIED FORCE

Select the **Direction** of the applied force — **Space-fixed direction** or **Body-fixed direction**.

- For **Space-fixed direction**, the directions of the applied force $\mathbf{F}$ are fixed with respect to the selected **Coordinate system**.
- For **Body-fixed direction**, the directions of the applied force $\mathbf{F}$ follow the rotation of the rigid domain.

Enter values or expressions for the components of the **Applied force** $\mathbf{F}$. The direction coordinate names can vary depending on the selected coordinate system.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Connector** > **Applied Force**
Shell > **Rigid Connector** > **Applied Force**
Beam > **Rigid Connector** > **Applied Force**
Multibody Dynamics > **Rigid Connector** > **Applied Force**

Ribbon

Physics tab with **Rigid Connector** node selected in the model tree:

Attributes > **Applied Force**

Applied Moment (Rigid Connector)

Use the **Applied Moment** subnode to apply a moment at the center of rotation. The moment that you specify is interpreted in the selected coordinate system.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model. Select a **Coordinate system** for specifying the directions of the moment.

In 2D, the moment is a scalar input, and is not affected by the coordinate system selection.

In order to be applicable, a coordinate system must have axis directions that are independent of the location in space.

APPLIED MOMENT

For 3D components, select the **Direction** of the applied moment — **Space-fixed direction** or **Body-fixed direction**.

- For **Space-fixed direction**, the directions of the applied moment $\mathbf{M}$ are fixed with respect to the selected **Coordinate system**.
- For **Body-fixed direction**, the directions of the applied moment $\mathbf{M}$ follow the rotation of the rigid domain.

Enter values or expressions for the **Applied moment** $\mathbf{M}$.

- For 3D components, enter the **x**, **y**, and **z** components of $\mathbf{M}$.
- For 2D components, enter the applied moment around the z-axis, M_z.

LINEAR BUCKLING

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Connector** > **Applied Moment**
Shell > **Rigid Connector** > **Applied Moment**
Beam > **Rigid Connector** > **Applied Moment**
Multibody Dynamics > **Rigid Connector** > **Applied Moment**

Ribbon

Physics tab with **Rigid Connector** node selected in the model tree:

Attributes > **Applied Moment**

Mass and Moment of Inertia (Rigid Connector)

Use the **Mass and Moment of Inertia** subnode to add inertia properties to the rigid connector for dynamic analysis.

CENTER OF MASS

Here you specify the location of the center of mass for the contribution given in this node. Select an option from the list: **Center of rotation** or **User defined**.

- For **Center of rotation**, the location of the load is taken as the center of rotation as defined in the parent **Rigid Connector** node.
- For **User defined**, enter the **Global coordinates of center of mass**, $\mathbf{X}_{\mathrm{m}}$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_{offset}.

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{m}} = \mathbf{X}_{\mathrm{m,input}} + \mathbf{X}_{\mathrm{offset}}$$

MASS AND MOMENT OF INERTIA

Enter values or expressions for the **Mass** m. Also enter values or expressions for the **Moment of inertia** $\mathbf{I}$. The axis directions of the moment of inertia tensor are given by the coordinate system selection in the parent Rigid Connector node.

- For 3D components, select **Isotropic**, **Diagonal**, or **Symmetric** and enter one or more components for the tensor $\mathbf{I}$.
- For 2D components, enter a value or an expression for I_z.

FRAME ACCELERATION FORCES

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Connector** > **Mass and Moment of Inertia**

Shell > **Rigid Connector** > **Mass and Moment of Inertia**

Beam > **Rigid Connector** > **Mass and Moment of Inertia**

Multibody Dynamics > **Rigid Connector** > **Mass and Moment of Inertia**

Ribbon
Physics tab with **Rigid Connector** node selected in the model tree:

Attributes > **Mass and Moment of Inertia**

Spring Foundation (Rigid Connector)

Add a **Spring Foundation** subnode to specify a spring or damper connecting the rigid connector to a fixed ground. The spring can act at an arbitrary position in space. A translational spring implicitly contributes also to the moment if it is not applied at the center of rotation of the rigid connector. The data that you specify is interpreted in the selected coordinate system.

LOCATION

Select an option from the list: **Center of rotation** or **User defined**. This is the location where the spring is attached.

- For **Center of rotation**, the spring will be attached to the center of rotation as defined in the parent **Rigid Connector** node.
- For **User defined**, enter the **Location (global coordinates)**, $\mathbf{X}_\mathrm{p}$, in the table.

Select the **Offset** checkbox to add an optional offset vector to the definition of the location. Enter values for the offset vector X_offset.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_\mathrm{P} = \mathbf{X}_\mathrm{P,input} + \mathbf{X}_\mathrm{offset}$$

SPRING

Select a **Spring type** — **Spring constant** or **Force as function of extension**.

When **Spring constant** is selected, the translational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Force as function of extension** is selected, enter the force vector $\mathbf{F}_\mathrm{s}$. It must be a function of the built-in variables describing the spring extension. The default value indicates the correct variable name, for example, `solid.rig1.uspring1_spf1`.

ROTATIONAL SPRING

Select a **Spring type** — **Spring constant** or **Moment as function of rotation**.

When **Spring constant** is selected, the rotational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

When **Moment as function of rotation** is selected, enter the moment vector $\mathbf{M}_s$. It must be a function of the built-in variables describing the spring extension. The default value indicates the correct variable name, for example, `solid.rig1.thspring1_spf1`.

In 2D the **Spring constant** is only a single scalar, representing the stiffness for rotation around the out-of-plane direction. Similarly, the **Moment as function of rotation** is a single scalar.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** $\eta_{u,s}$, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions in the table for the **Loss factor for spring** $\eta_{u,k}$ or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

All settings in this section are analogous to the corresponding settings in the Loss Factor Damping section. In 2D, only one scalar loss factor, corresponding to rotation around the out-of-plane axis, is given.

VISCOUS DAMPING

Select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions for the damping constants d_u in the table. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS ROTATIONAL DAMPING

All settings in this section are analogous to the corresponding settings in the Viscous Damping section. In 2D, only one scalar viscous damping, corresponding to rotation around the out-of-plane axis, is given.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Rigid Connector** > **Spring Foundation**
Multibody Dynamics > **Rigid Connector** > **Spring Foundation**
Shell > **Rigid Connector** > **Spring Foundation**
Beam > **Rigid Connector** > **Spring Foundation**

Ribbon

Physics tab with **Rigid Connector** node selected in the model tree:

Attributes > **Spring Foundation**

Attachment

The **Attachment** node is used to define a set of boundaries on a flexible domain which can be used to connect it with other components through a joint in the Multibody Dynamics interface.

There are two formulations of the attachment. In the default formulation, all the selected boundaries behave as if they were connected by a common rigid body. In some cases, this give an unwanted stiffening or unrealistic local stresses. You can then switch to a flexible formulation where the constraint is applied only in an average sense.

Using attachments is optional when connecting a rigid component. It is, however, useful if you consider switching between a rigid and a flexible representation, since you then do not have to change the settings in the joints.

Attachments can be added to boundaries in a Multibody Dynamics or a Solid Mechanics interface, to edges in a Shell or Layered Shell interface, or to points in a Beam interface. This makes it possible to use a joint in the Multibody Dynamics interface for connecting parts modeled in different physics interfaces.

The use of attachments is discussed in the Attachments section in the *Structural Mechanics Modeling* chapter.

The **Attachment** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available in 3D, 2D, and 2D axisymmetry.

CONNECTION TYPE

Select **Rigid** or **Flexible**.

When the connection type is rigid, the whole attachment selection acts as a virtual rigid object. This will cause a stiffening of the part with the attachment, and local stress concentrations at the selected boundaries if the domain with the attachment is flexible.

In the flexible formulation, the selected boundaries are allowed to have internal deformations, and the kinematic constraints are fulfilled only in an average sense. This will essentially provide a statically equivalent load distribution over the boundaries.

If a selected boundary is located on a rigid domain, the connection type setting does not matter. The rigid formulation is always used.

ADVANCED

Select **Group dependent variables in solver** — **From physics interface** (default), **Yes**, or **No**, to choose how to group in the solver sequence the dependent variables added by the **Attachment** feature.

With the flexible formulation, some extra degrees of freedom are added for each attachment. In order to get good convergence properties for a nonlinear solution, these variables must have a good scaling.

When a new solver sequence is generated, then an appropriate manual scaling is automatically set for these variables.

If, however, an existing solver sequence was generated while **Connection Type** was set to **Rigid**, and you then change to **Flexible**, no such scaling will be present. In this case, you either have to regenerate the solver sequence, or set the scaling manually under the **Dependent Variables** node in each study step.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Connections** > **Attachment**

Ribbon
Physics tab with **Solid Mechanics** selected:

Boundaries > **Connections** > **Attachment**

Thermal Expansion (Attachment)

Add the **Thermal Expansion** subnode to prescribe a deformation of the attachment caused by changes in temperature. This makes it possible to reduce stresses caused by the attachment being rigid, while there are thermal deformations in the flexible body to which it is attached.

[Select an **Input type** to specify how the thermal strain is entered. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

here, the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\text{t}}(\tau)d\tau\right) - 1$$

where α_{t} is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain in the attachment.

Select **Inherit from domain** to take the thermal expansion data from the domain to which it is attached. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual material in the attachment has the same thermal expansion as the domain itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for the **Volume reference temperature** T_{ref} that is the temperature at which there are no thermal displacements at the constraints.
- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any temperature variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The default values **From material** are used. This requires that a material has been assigned to the boundaries, edges, or points where the constraint is active. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the *Structural Mechanics Modeling* chapter.
- Thermal Expansion of Constraints in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Attachment** > **Thermal Expansion**

Multibody Dynamics > **Attachment** > **Thermal Expansion**

Ribbon

Physics tab with **Attachment** node selected in the model tree:

Attributes > **Thermal Expansion**

Reduced Flexible Components

Add a **Reduced Flexible Components** node to define one or more flexible parts of the structure as a reduced-order model (ROM) for component mode synthesis (CMS).

The **Reduced Flexible Components** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available in 3D and 2D.

COMPONENT MODE SYNTHESIS

Select a **Component Definition** — **Detect from geometry** or **User defined**. If **Detect from geometry** is used, then a geometric analysis is performed, and the selection is subdivided into a number of disjoint components. For each such component found, a *Component Definition* subnode is added. In order to do manual selection of the domains to be included in the reduced components, select **User defined**.

Enter a required **Number of eigenmodes** to be included in the dynamic representation of each reduced component. This value can be modified for individual components in the settings in the **Component Definition** nodes.

When the settings in this node and its Component Definition subnodes are ready, click the **Automated Model Setup** icon ( ) on the **Component Mode Synthesis** section toolbar and select the **Configure CMS Study** ( ) button to set up the study sequence that generates the ROMs for all selected components.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Include load contributions in Reduced Components** checkbox if any loads in the physics interface should be part of the reduced component. A load that is included in a reduced component will act when that reduced component is used in a global study. In order to avoid double contributions, that same load must then be disabled when running the global study. The default is such that this checkbox is cleared, in which case loads are not included in the reduced components and will be only active when running the global study.

When **Synchronize 'Solve for' study setting for Reduced Components** is selected, the **Solve for** column in the table in the **Physics and Variables Selection** section in applicable study steps is automatically managed for consistency between ROMs and their generating physics interface. To take manual control, clear this checkbox.

When **Include connections and pairs in component definition** is selected, geometrically disconnected domains that are connected by pair features or connection features in the Shell interface are merged. Such features include:

- **Continuity**
- **Thin elastic layer**
- **Boundary to Boundary**
- **Edge to Boundary**
- **Edge to Edge**.

See also

- Component Mode Synthesis in the *Structural Mechanics Modeling* chapter.
- Model Reduction in the *COMSOL Multiphysics Reference Manual.*

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Reduced Flexible Components**
Shell > **More** > **Reduced Flexible Components**
Multibody Dynamics > **More** > **Reduced Flexible Components**

Ribbon

Physics tab with **Solid Mechanics, Shell**, or **Multibody Dynamics** selected:

Domains > **More** > **Reduced Flexible Components**

Component Definition

The **Component Definition** subnode under Reduced Flexible Components is used for selecting the domains that form a reduced component. By default, a number of **Component Definition** nodes are automatically generated when a selection is made in the parent node. Each **Component Definition** node then contains a selection of domains that form a contiguous object, based on an analysis of the geometry.

You can modify the initial automatic selections, and add or remove **Component Definition** nodes as needed. To make manual selections possible, you must set **Component Definition** to **User defined** in the parent node.

The **Component Definition** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COMPONENT MODE SYNTHESIS

Select the **Number of eigenmodes** to be used when building this reduced component. The default is **From parent**. Select **User defined** to enter another value.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Reduced Flexible Components** > **Component Definition**
Shell > **Reduced Flexible Components** > **Component Definition**
Multibody Dynamics > **Reduced Flexible Components** > **Component Definition**

Ribbon

Physics tab with **Reduced Flexible Components** node selected in the model tree:

Attributes > **Component Definition**

Fixed Joint

The **Fixed Joint** node is used to prevent any relative motion between two components, but it can also be used for connecting reduced-order models in the context of Component Mode Synthesis.

The components connected by a **Fixed Joint** are rigidly attached to each other, and the forces or moments required to restrict the relative motion can be computed. A **Fixed Joint** node connects, for example, two attachments so that they form a common rigid region.

- If a Multibody Dynamics Module license is available, you can get access to many different joint types by adding a Multibody Dynamics interface or a Joints interface to your model.
- In contexts other than component mode synthesis, connecting two parts by a Rigid Connector is usually a more natural approach.

The **Fixed Joint** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

ATTACHMENT SELECTION

Select a **Source** and a **Destination** attachment from the lists. You can select any Attachment or Rigid Material in the model. In case that there is a Multibody Dynamics interface in the model, also such features as **Spur Gear**, **Helical Gear**, **Bevel Gear**, **Worm Gear, Spur Rack**, and **Helical Rack** can be selected. In **Source**, an additional option, **Fixed**, is also available. When **Fixed** is selected, the source side of the joint is fixed in space, and not connected to any modeled parts. If any **Base Motion** nodes are present in a Multibody Dynamics interface, these also appear in the list for the **Source** attachment. Selecting a base motion gives a behavior similar to using **Fixed**, with the exception that the base now can have a nonzero prescribed displacement, velocity, or acceleration.

In large models, the list of available attachments can become long. You can then create geometric filters to narrow down the search. When the **Use selection filter** checkbox is selected, two subnodes named Source Filter and Destination Filter are added to the joint node. In these subnodes you can make graphic selections of the objects whose attachments should be shown in the **Source** and a **Destination** lists.

JOINT FORCES AND MOMENTS

Select an option from the list to use the method of computation for the forces and moments — **Do not compute**, **Computed using weak constraints**, or **Computed using penalty method**.

For **Computed using penalty method**, enter the **Penalty factor** p_j. The default is `solid.fxj1.Eequ*0.1*solid.diag`. The exact naming of the variable depends on the tags of the nodes in the model tree.

Computing Joint Forces and Moments

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Fixed Joint**
Shell > **Fixed Joint**

Ribbon

Physics tab with **Solid Mechanics** or **Shell** selected in the Model Builder tree:

Global > **Solid Mechanics** > **Fixed Joint**

Source Filter

When you select domains in this node, the **Source** list in the parent **Fixed Joint** node will only contain attachments and rigid domains that are part of the selection. The selection rules are:

- If any part of an active rigid domain is within the selection, that rigid domain will appear in the list.
- If any boundary of an active attachment is adjacent to the selected domains, that attachment will appear in the list.
- If the selection is such that no attachments or rigid domains satisfy the rules above, the **Source** list in the parent node will only contain the entry **None**.
- If the selection is such that some attachments or rigid domains satisfy the rules above, the **None** entry in the **Source** list in the parent node will be removed, and one of the attachments or rigid domains will be selected as default. This means that if the selection is such that the **Source** list has only one unique item, this item is automatically selected.
- If the selection is empty, the behavior is as if **Use selection filter** were not selected.

LOCATION IN USER INTERFACE

This node is automatically created when the **Use selection filter** checkbox is selected in the **Attachment Selection** section in the parent **Fixed Joint** node. The **Source Filter** node cannot be added or removed manually.

Destination Filter

When you select domains in this node, the **Destination** list in the parent **Fixed Joint** node will only contain attachments and rigid domains which are part of the selection. The selection rules are described in Source Filter.

LOCATION IN USER INTERFACE

This node is automatically created when the **Use selection filter** checkbox is selected in the **Attachment Selection** section in the parent **Fixed Joint** node.The **Destination Filter** node cannot be added or removed manually.

Contact

In the **Contact** node, you define the mechanical and numerical properties for a set of contact pairs in a contact analysis. Use it for modeling structural contact and multiphysics contact. In the latter case, you will also need to add corresponding pair conditions in the other participating physics interfaces.

In the Solid Mechanics interface, you have also the option to use a simplified contact formulation on interior boundaries, Interior Contact.

In the contact pair, the destination side selection must be part of the physics interface in which you add the **Contact** node. The source side selection can be any kind of meshed object, but if it is part of the current physics interface, more options are made available. It is also possible to use the same selection for both the destination and source side to, for example, model self-contact. Note that the source and destination may not partially intersect each other.

- If friction is to be included in the modeling, add a Friction or a Slip Velocity subnode. The latter provides a simplified modeling approach, when the state of friction and the direction of slip between the two boundaries are known.
- In models where contact is not established in the initial configuration, a singularity may occur. In order to suppress such singularities, you can add a Stabilization subnode.
- You can compute the wear caused by sliding by adding a Wear subnode.
- You can also model perfect bonding and debonding of the contact pair using the Adhesion and Decohesion subnodes.

The **Wear** subnode is not available in the Layered Shell interface.

If contact pairs are present and active in the model component, a default **Contact** node is added to the physics with all contact pairs selected. Contact pairs selected in a **Contact** node are exclusive, which means that if a contact pair is selected more than once, only its last occurrence in the physics tree will be active. Hence, by adding new **Contact** nodes, physics properties and settings related to specific contact pairs can be set.

Moreover, contact equations are only added if and only if a destination boundary of an active contact pair intersects with the selection of the physics interface. If only source boundaries are active, it is, however, possible to for example add offsets to the contact

surface. This can be useful when setting up contact conditions across multiple physics interfaces.

Which of the **Contact** settings described below are visible, depends on the applicable selections of the specific feature.

- If both source and destination boundaries are applicable, all settings are visible.
- If only source boundaries are applicable, only settings that affect variables on the source boundaries are visible.
- If only destination boundaries are applicable, only settings that affect equations and variables on the destination boundaries are visible.
- If no boundaries are applicable, no settings are visible.

If no contact pairs are selected, all settings are visible.

Updating the settings window of the **Contact** node based on the applicable selections can result in some delay for models with a large number of pairs. This functionality can be turned off by disabling the **Selection Information** ( ) option in the **Show More Options** dialog ( ), which may improve the responsiveness of the user interface.

The **Contact** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

In order to specify contact conditions, one or more **Contact Pair** nodes must be available in the **Definitions** branch.

If you have several physics interfaces with displacement degrees of freedom on the same geometrical objects in your model, there are special considerations for contact modeling. The gap and slip are determined by the locations of the contacting objects in the spatial frame. This means that you can only model contact between physics interfaces that either control the spatial frame, or have displacements which are identical to such an interface.

When a **Contact** node is present in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** checkbox on the study step **Settings** window is selected and cannot be cleared.

For the default **Contact** node, selecting **Disconnect pair** in the Contact method section removes the above requirement and makes it possible to clear the **Include geometric nonlinearity** checkbox.

When at least one **Contact**, **General Contact**, or **Interior Contact** node is active in the model, a **Contact Force** plot group is added to the list of result templates. The plot group contains one vector plot showing the contact pressure for each contact node. If a **Friction** node is present, then there will also be a vector plot of the tractions caused by friction.

- Contact Modeling
- Contact Analysis Theory

CONTACT SURFACE

When **Contact** is used in a Layered Shell, Shell, or Membrane interface, settings that control which side of the boundary that can come into contact are also made available.

For **Contact surface, destination** and **Contact surface, source**, select **Top** or **Bottom**. The top side is the one with an outward pointing normal vector.

This section is only present when **Contact** is used in a:

- Layered Shell interface
- Shell interface
- Membrane interface

For a longer discussion about controlling and visualizing normal vectors, see Contact Detection

CONTACT METHOD

Select the algorithm used for computing the contact, **Penalty** (default); **Penalty, dynamic**; **Augmented Lagrangian**; **Augmented Lagrangian, dynamic**; or **Nitsche**. The augmented Lagrangian method is in general more accurate than the penalty method, but also more expensive in terms of computer resources. The Nitsche method is also a more accurate method that is available with it the Solid Mechanics and Multibody Dynamics interfaces.

The options **Augmented Lagrangian** and **Augmented Lagrangian, dynamic** are not available in the Solid Mechanics, Explicit Dynamics interface.

When **Augmented Lagrangian** is selected, it is also possible to choose **Solution method** — **Segregated** (default) or **Fully coupled**. This setting affects how the default solver suggestion is set up when generating a new solver sequence, as well as some variable definitions and equations.

The **Penalty, dynamic**; and **Augmented Lagrangian, dynamic** formulations are intended for use in time-dependent studies to model dynamics contact, such as soft impact type events.

The **Nitsche** method can conceptually be seen as an enhancement of the penalty method where the surface traction of the adjacent domains is utilized to improve accuracy. When selected, it is also possible to choose a **Formulation** — **Symmetric**, **Skew-symmetric**, or **Incomplete** (default). This setting controls how the equations added to implement the contact condition are written and can influence the robustness and performance of the solution. The default **Incomplete** formulation is the least expensive of the three, and, in the majority of cases, gives an accurate and robust solution.

For the default **Contact** node, it is possible to disconnect the selected contact pairs by selecting **Disconnect pair**. When the pairs are disconnected, they will not add contributions by adding equations or variables, but the region is still identified as a pair region. Also, the presence of the **Contact** node will require the study to be geometrically nonlinear.

CONTACT PRESSURE PENALTY FACTOR

The settings in this section are used to control the penalty factor used by the pressure contact model. The settings available depend on the chosen contact method.

In the penalty method, the penalty factor is the actual stiffness of a spring inserted between the boundaries in the contact pair.

In the augmented Lagrangian method, the penalty factor controls how "hard" the interface surface is during the iterations, but it does not affect the converged results. You can consider the penalty factor as a spring giving a resisting force if the boundaries in the contact pair have an overclosure. For a segregated solution, a larger value gives faster but less stable convergence to the state where there is no overclosure. A too large value can create convergence difficulties, particularly if the overclosure is large. For a coupled solution, the penalty factor mainly affects the structure of the underlying equation, which can alter the convergence properties of the model.

In the Nitsche method, the penalty factor is mainly a stabilization parameter and, in theory, it does not affect the accuracy of the solution. However, using a too low penalty factor causes instability which is detrimental to the accuracy of the solution.

Settings for the Penalty Method

Select the type of **Penalty factor control** — **Automatic**; **Automatic, soft**; **Manual tuning**; **Nonlinear**; or **User defined**.

- **Automatic** and **Automatic, soft** provide predefined values for the penalty factor multiplier f_{p}. The **Automatic, soft** is mainly intended for bending dominated problems, where the stiffness of the contacting structures are not directly controlled by compression normal to the contact surface. This is typically encountered when using contact with the Shell interface.
- For **Manual Tuning**, enter a **Penalty factor multiplier** f_{p}. The default value is 1, which corresponds to the **Automatic** option.
- For **Nonlinear**, enter an **Initial penalty factor multiplier** $f_{\mathrm{p,init}}$ and a **Final penalty factor multiplier** $f_{\mathrm{p,fin}}$. Enter a value for the **Transition gap** $g_{\mathrm{n,tr}}$ to control when to switch

between the initial and final penalty factors. The default value is `-<phys>.hmin_dst*1e-3`.

- For **User defined**, enter a **Contact pressure penalty factor** p_n. The default value is `<phys>.<contact_tag>.E_char/<phys>.hmin_dst`. The variable `E_char` contains the value given as characteristic stiffness, and `hmin_dst` is the minimum element size on the destination selection.

Select the type of **Penalty function** — **Ramp** or **Smooth ramp**.

Click to select **Offset penalty function** if you want the contact pressure to be nonzero when the gap is zero. For the **Ramp** function, enter a value for the **Contact pressure at zero gap** T_0. For the **Smooth ramp** function, enter the **Start location** Δ_{start} and the **Transition zone size** Δ_{size}.

Through this option, you can compensate for the overclosure that is inherent to the penalty method if you know an approximate value of the expected contact pressure.

Settings for the Penalty, Dynamic Method

Select the type of **Penalty factor control** — **Automatic**; **Automatic, soft**; **Manual tuning**; **Nonlinear**; **Viscous only**; or **User defined**. The settings are the same as for the standard penalty method, except for the **Viscous only** option. By selecting **Viscous only**, the stiffness terms of the penalty contact are omitted from the formulation.

Select the type of **Viscous penalty factor control** — **Automatic** or **User defined**. For **Automatic**, enter a value for the **Characteristic time** τ_n. This value can be used as a multiplier for the viscous penalty factor, but should as a rule-of-thumb be in the same order of magnitude as the duration of the contact event. When **User defined** is selected, enter the **Viscous contact pressure penalty factor** p_{nv}. The default value is `(<phys>.<contact_tag>.E_char/<phys>.hmin_dst)*1`[ms]. The variable `E_char` contains the value given as characteristic stiffness, and `hmin_dst` is the minimum element size on the destination selection.

Settings for the Segregated Augmented Lagrangian Method

Select the type of **Penalty factor control** — **Preset** (default), **Manual tuning**, or **User defined**. The settings give access to an increasing level of detailed control of the penalty factor, and are available when **Solution method** is set to **Segregated**.

If you select **Preset**, you have the options to select **Tuned for** to be **Stability** (default), **Speed**, or **Bending**. If the contact boundaries move toward each other, so that large overclosures can be expected in the initial outer iterations, then **Stability** is the better choice since it relaxes the penalty factor during the first iterations. In many models, where the contact state does not change much, using **Speed** gives significant

performance improvements. The **Bending** option provides a low penalty factor equal to the one used for the first iterations when using **Stability**. This option thus provides a conservative algorithm that can be particularly useful for problems dominated by bending. However, it might require many iterations to reach convergence.

The **Penalty factor control** selection **Manual tuning** gives you access to a number of detailed settings for the penalty factor.

Enter a **Penalty factor multiplier** f_{p}. The default value is 1. From **Use relaxation**, select **Always** (default), **Never**, or **Conditional**. When using relaxation, the penalty factor is decreased during the first outer iterations in each parameter or time step. The purpose is to avoid large residuals in the case of a large initial overclosure between the contacting boundaries.

If relaxation is used, enter the **Initial Relaxation Factor** r_{i}. The default is 0.005. This factor multiplies the penalty factor in the first outer iteration. Enter **Number of iterations with relaxation** N_{r}. The default value is 4. The penalty factor is gradually increased up to its full value, which is used in the outer iteration after the one where the specified number of iterations with relaxation have been reached.

If **Use relaxation** is set to **Conditional**, enter a **Suppression criterion**. The default is 0, which means that the relaxation is used for all parameter or time steps. This is a Boolean expression which, when fulfilled, suppresses the relaxation. If you, for example, know that your problem needs relaxation only during the initial phase of the solution, enter an expression like `load_parameter>0.1`. It is also possible to use expressions based on the solution, for example `abs(solid.cnt1.gap)<0.05*h`. This expression is true when the gap or overclosure is small compared to the mesh size. It could be taken as an indication that the contact problem is almost converged, and thus not in need of any relaxation of the penalty factor.

Using the **Penalty factor control** selection **User defined** gives you the possibility to enter an explicit expression for penalty factor. Enter the **Contact pressure penalty factor** p_{n}. The default is `(min(1e-3*(5^niterCMP,1))*`*`<phys>`*`.<contact_tag>`*`.E_char)/`*`<phys>`*`.hmin_dst`. The default value causes the penalty factor to be increased during the outer iterations and takes material stiffness and element size at the contact surface

into account. The variable `E_char` contains the value given as characteristic stiffness, and `hmin_dst` is the minimum element size on the destination selection.

The default values when **Manual tuning** is selected corresponds to the default expression when **User defined** is selected. The same expression is also used internally when **Preset** is used with the **Stability** option.

Settings for the Fully Coupled Augmented Lagrangian Method and the Nitsche Method

Select the type of **Penalty factor control** — **Automatic** (default), **Manual tuning**, or **User defined**.

- For **Manual Tuning**, enter a **Penalty factor multiplier** f_{p}. The default value is 1, which corresponds to the **Automatic** option.
- For **User defined**, enter a **Contact pressure penalty factor** p_{n}. The default value is *<phys>*`.`*<contact_tag>*`.E_char/`*<phys>*`.hmin_dst`. The variable `E_char` contains the value given as characteristic stiffness, and `hmin_dst` is the minimum element size on the destination selection.

Settings for the Augmented Lagrangian, Dynamic Method

Select the type of **Viscous penalty factor control** — **Preset** (default), **Manual tuning**, or **User defined**. The settings give access to an increasing level of detailed control of the penalty factor.

When **Preset** or **Manual Tuning** is selected, enter a value for the **Characteristic time** τ_{n}. This value can be used as a multiplier for the viscous penalty factor, but should as a rule-of-thumb be in the same order of magnitude as the duration of the contact event. The same settings as for the segregated augmented Lagrangian method are available for the **Preset** and **Manual Tuning** options. However, the default is to use no relaxation when the **Augmented Lagrangian, dynamic** method is used.

Using the **Penalty factor control** selection **User defined** gives you the possibility to enter an explicit expression for penalty factor. Enter the **Viscous contact pressure penalty factor** p_{nv}. The default value is `(`*<phys>*`.`*<contact_tag>*`.E_char/`*<phys>*`.hmin_dst)*l`[ms]. The variable `E_char` contains the value given as characteristic stiffness, and `hmin_dst` is the minimum element size on the destination selection.

TRIGGER CUTBACK

Select the **Trigger cutback** checkbox to enable additional control over solver cutbacks in a time-dependent study or a stationary study with an auxiliary sweep. Enter a logical expression for **Cutback criterion**. When this expression evaluates to a nonzero value, the iterations are immediately terminated, and the solver tries to use a smaller value of the time or parameter. You can use this setting to avoid that the solver spends many iterations trying to recover from an unphysical state. As an example, if you know that the destination boundary is confined, so that it will not move more than 1 mm anywhere, an entry like `solid.disp > 5[mm]` can be useful, since it is unlikely that a displacement of this size should appear during a successful iteration history.

In the *COMSOL Multiphysics Reference Manual:*

- Time Dependent and Stationary
- About the Time-Dependent Solver and About the Parametric Solver

CONTACT SURFACE OFFSET AND ADJUSTMENT

Enter a value or an expression for **Offset from geometric destination surface** $d_{\text{offset,d}}$. The offset is subtracted from the gap in the normal direction of the destination surface.

Enter a value or an expression for **Offset from geometric source surface** $d_{\text{offset,s}}$. The offset is subtracted from the gap in the normal direction of the source surface.

Select **Force zero initial gap** to compensate for any difference caused by irregularities in geometry or mesh when the two contacting boundaries should exactly coincide in the initial state. Gaps smaller than the tolerance Δ_{gap} are adjusted to be zero before any

offset is added. By default, Δ_{gap} is set to `Inf`, meaning that all gaps detected are adjusted to be zero.

Use the offset properties to adjust initial clearances (negative values) or interference fits (positive values) without having to change the geometry. These properties are also useful for studying the effects of geometrical tolerance when the structure is still modeled using its nominal size.

When combined with **Force zero initial gap**, the offset will be exact in the sense that it is not affected by mesh irregularities.

The adjustment made by **Force zero initial gap** does not move any nodes in the mesh. Effectively, it adds an extra hidden offset, which compensates for the initial distance between source and destination boundaries.

Contact Surface Offset and Adjustment

INITIAL VALUE

This section is only available when **Formulation** is **Augmented Lagrangian** or **Augmented Lagrangian, dynamic**. In the augmented Lagrangian method, the contact pressure is introduced as an extra dependent variable on the destination boundaries. Enter a value for **Contact pressure** T_n to supply an initial value for the contact pressure.

- It is possible to speed up the convergence by supplying a guess of the correct order of magnitude of the contact pressure.
- If not all the contacting parts are fully constrained, it is sometimes possible to avoid singular problems by introducing a nonzero value for the contact pressure.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

This section is only available when **Formulation** is **Augmented Lagrangian** or **Augmented Lagrangian, dynamic**. In the augmented Lagrangian method, the contact pressure is introduced as an extra dependent variable on the destination boundaries. You can

modify the shape function type and order of this field. The default is to always use a linear shape function order for the contact pressure, but using a higher order can improve the accuracy of the contact condition. A value up to the discretization order of the displacement field is allowed. This setting also affects the shape function for dependent variables added by any subnode added to the **Contact** node.

To change the discretization, select **Shape function type** — **Lagrange** or **Nodal serendipity**.

- For the **Lagrange** shape-function type, select **Element order** — **Linear**, **Quadratic**, **Cubic**, **Quartic**, or (in 2D) **Quintic**.
- For the **Nodal serendipity** shape-function type, select **Element order** — **Quadratic**, **Cubic**, or **Quartic**. The **Nodal serendipity** option is not available when the order of the displacement field is linear.

The lumped solver used for the segregated augmented Lagrangian method is only optimal when the shape function order is **Linear**; otherwise, use a standard segregated step instead. The proper solver sequence is set up when adding a new default solver.

QUADRATURE SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Typically, the expressions in the contact weak equations include discontinuous functions. It is sometimes preferable to use a high integration order to improve the accuracy of the numerical integration of such expressions. By default, the numerical integration of the contact weak equations is twice the shape function order of the displacement field. Clear the **Use automatic quadrature settings** checkbox to specify a custom order. When the checkbox is cleared, enter an integer value between 0 and 41 in the **Integration order** field. This setting also affects the integration order for any subnode added to the **Contact** node.

- The **Integration order** affects the numerical integration of the contact weak contributions as well as the number of state variables used when, for example, computing friction forces.
- Increasing the integration order increases the time spent on assembling the system matrices each iteration. This increase can be significant if a too high order is used since the contact mapping is done at each integration point on the destination boundaries.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Enter a **Characteristic stiffness** E_{char}. The default is *<phys>*`.Eequ`, the equivalent Young's modulus as defined by most material models. The characteristic stiffness (stored in a variable named *<phys>*`.`*<contact_tag>*`.E_char`) is used in expressions for the default penalty factors for both the penalty and the augmented Lagrangian methods. The characteristic stiffness should be representative for the stiffness of the destination domain material in a direction normal to the boundary. You may for example need to adjust the default value in for the following cases:

- The material is strongly anisotropic. The default value is based on an average in all directions.
- The material is highly nonlinear. The default value is based on the stiffness at zero strain.
- The variable *<phys>*`.Eequ` is not defined by the material. This is the case for some user-defined materials.

Multiphysics Contact Detection

When analyzing a multiphysics contact problem, the state of being in contact or not will be passed from the contact analysis in the structural mechanics interface to other participating physics interfaces. For numerical reasons, the gap value will however not be exactly zero, even when the boundaries are in contact. A certain small positive value of the gap will thus be considered as being in contact. The default **Multiphysics contact tolerance** is **Automatic**. If you want to explicitly specify the limit of the gap considered as being in contact, select **Manual**, and enter the **Contact tolerance** Δ_{contact}.

The automatic multiphysics contact detection tolerance is based on the size of the geometric entity.

Jacobian Contribution

The equations for the contact conditions can be implemented differently in order to influence the robustness and stability of the discretized system of equations and its linearization. This can be controlled by the **Jacobian Contribution** setting.

When the **Jacobian Contribution** is set to **Nonsymmetric**, the variations of the normal vector on the source boundary are excluded. This is the most robust implementation for cases where the source boundary undergoes large deformation. However, it has the

drawback of causing the global stiffness matrix to become nonsymmetric, which increases the memory requirement in the linear solver.

The **Symmetric** option can be used instead when this is problematic for the linear solver. For this option, the main contact equations does not force a nonsymmetric stiffness matrix, although some subnodes may still do so. Variations of the normal vector on the source boundary are excluded also for this option, but it is not as robust as the **Nonsymmetric** option.

When the **Jacobian Contribution** is set to **Automatic**, the **Nonsymmetric** option is used for contact pairs using the **Deformed configuration** mapping method, while the **Symmetric** option is used for the **Initial configuration** mapping method.

When the **Jacobian Contribution** is set to **Legacy**, variations of the normal vector on the source boundary are included, and the contribution to the global stiffness matrix is symmetric.

For the **Nitsche** method, the **Jacobian Contribution** setting is not available, since the equations are always formulated according to the **Nonsymmetric** option.

Fields Excluded from Variation

In some cases, it is not desirable to include reaction forces from all physics when setting up the variational term of the contact weak contributions. When **Fields excluded from variation** is set to **Automatic**, variations are taken with respect all fields expect when a **Wear** subnode is added. If a **Wear** subnode is active and uses a **Deformed geometry** formulation, variations with respect to the material mesh displacement field is excluded in all weak equations added by **Contact** and its subnodes. By setting **Fields excluded from variation** to **Manual**, it is possible to exclude variations with respect to any field present in the model. When **Manual** is selected, enter a comma separated list of field names to exclude. By default, the input field is populated with `material.disp`, which exclude variations with respect to material mesh displacement field added by the deformed geometry functionality.

Select **Add contact status to solver log** to get printouts about the change in the contact state in the solver log window. Doing so will add internal dependent variables used for tracking the contact state on the destination boundaries.

The **Penalty, dynamic** and **Augmented Lagrangian, dynamic** are dissipative formulations. When either of these are selected in the Contact Method section, select **Compute viscous contact dissipation** to compute and store to the energy dissipated by the viscous

pressure contact. This adds one extra dependent variable and an extra distributed ODE.

Dependent Variables in Contact Analysis

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Pairs** > **Contact**
Solid Mechanics, Explicit Dynamics > **Pairs** > **Contact**
Shell > **Pairs** > **Contact**
Layered Shell > **Pairs** > **Contact**
Membrane > **Pairs** > **Contact**
Multibody Dynamics > **Pairs** > **Contact**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; **Layered Shell**; **Shell**; **Membrane**; or **Multibody Dynamics** selected:

Pairs > **Contact**

Interior Contact

In the **Interior Contact** node, you define contact conditions for a set of interior boundaries in a solid model. This contact formulation provides a simplified setup that does not require the addition of contact pairs. It is limited so small-scale sliding between the contact boundaries.

- If friction is to be included in the modeling, add a Friction subnode.
- You can also model perfect bonding and debonding, using the Adhesion and Decohesion subnodes.

The **Interior Contact** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

When at least one **Contact**, **General Contact**, or **Interior Contact** node is active in the model, a **Contact Force** plot group is added to the list of result templates. The plot group contains one vector plot showing the contact pressure for each contact node. If a **Friction** node is present, then there will also be a vector plot of the tractions caused by friction.

- Contact Modeling
- When to Use the Interior Contact Formulation
- Contact Analysis Theory

COORDINATE SYSTEM SELECTION

The local directions of normal and tangential directions for displacements and tractions for this node and any subnodes are determined by the selected coordinate system. The selection is limited to boundary systems.

CONTACT METHOD

Select the algorithm used for computing the contact, **Penalty** (default), **Augmented Lagrangian**, or **Nitsche**. The augmented Lagrangian method is in general more accurate than the penalty method, but also more expensive in terms of computer resources. The Nitsche method is also a more accurate method.

The option **Augmented Lagrangian** is not available in the Solid Mechanics, Explicit Dynamics interface.

When **Augmented Lagrangian** is selected, it is also possible to choose **Solution method** — **Segregated** (default) or **Fully coupled**. This setting affects how the default solver suggestion is set up when generating a new solver sequence, as well as some variable definitions and equations.

The **Nitsche** method can conceptually be seen as an enhancement of the penalty method where the surface traction of the adjacent domains is utilized to improve accuracy. For **Interior Contact**, the only available **Formulation** is **Incomplete**.

CONTACT PRESSURE PENALTY FACTOR

The settings in this section are used to control the penalty factor used by the pressure contact model. The settings available depend on the chosen contact method.

In the penalty method, the penalty factor is the actual stiffness of a spring inserted between the contacting boundaries.

In the augmented Lagrangian method, the penalty factor controls how "hard" the interface surface is during the iterations, but it does not affect the converged results. You can consider the penalty factor as a spring giving a resisting force if the boundaries have an overclosure. For a segregated solution, a larger value gives faster but less stable convergence to the state where there is no overclosure. A too large value can create convergence difficulties, particularly if the overclosure is large. For a coupled solution, the penalty factor mainly affects the structure of the underlying equation, which can alter the convergence properties of the model.

In the Nitsche method, the penalty factor is mainly a stabilization parameter and, in theory, it does not affect the accuracy of the solution. However, using a too low penalty factor causes instability which is detrimental to the accuracy of the solution.

Settings for the Penalty Method

Select the type of **Penalty factor control** — **Automatic**; **Automatic, soft**; **Manual tuning**; **Nonlinear**; or **User defined**.

- **Automatic** and **Automatic, soft** provide predefined values for the penalty factor multiplier f_{p}. The **Automatic, soft** is mainly intended for bending dominated problems, where the stiffness of the contacting structures are not directly controlled by compression normal to the contact surface.
- For **Manual Tuning**, enter a **Penalty factor multiplier** f_{p}. The default value is 1, which corresponds to the **Automatic** option.
- For **Nonlinear**, enter an **Initial penalty factor multiplier** $f_{\mathrm{p,init}}$ and a **Final penalty factor multiplier** $f_{\mathrm{p,fin}}$. Also enter a value for the **Transition gap**, $g_{\mathrm{n,tr}}$ to control when to switch between the initial and final penalty factors. The default value is `-<phys>.hmin_dst.*1e-3.`
- For **User defined**, enter a **Contact pressure penalty factor** p_{n}. The default value is *<phys>*`.`*<contact_tag>*`.E_char/`*<phys>*`.`*<contact_tag>*`.hmin`. The variable

`E_char` contains the value given as characteristic stiffness, and `hmin` is the minimum element size on the selected boundaries.

Click to select **Offset penalty function** if you want the contact pressure to be nonzero when the gap is zero. In that case, also enter a value for the **Contact pressure at zero gap** T_0. Through this option, you can to some extent compensate for the overclosure that is inherent to the penalty method if you know an approximate value of the expected contact pressure.

Settings for the Segregated Augmented Lagrangian Method

Select the type of **Penalty factor control** — **Preset** (default), **Manual tuning**, or **User defined**. The settings give access to an increasing level of detailed control of the penalty factor, and are available when **Solution method** is set to **Segregated**.

If you select **Preset**, you have the options to select **Tuned for** to be **Stability** (default), **Speed**, or **Bending**. If the contact boundaries move toward each other, so that large overclosures can be expected in the initial outer iterations, then **Stability** is the better choice since it relaxes the penalty factor during the first iterations. In many models, where the contact state does not change much, using **Speed** gives significant performance improvements. The **Bending** option provides a low penalty factor equal to the one used for the first iterations when using **Stability**. This option thus provides a conservative algorithm that can be particularly useful for problems dominated by bending. However, it might require many iterations to reach convergence.

The **Penalty factor control** selection **Manual tuning** gives you access to a number of detailed settings for the penalty factor.

Enter a **Penalty factor multiplier** f_p. The default value is 1. From **Use relaxation**, select **Always** (default), **Never**, or **Conditional**. When using relaxation, the penalty factor is decreased during the first outer iterations in each parameter or time step. The purpose is to avoid large residuals in the case of a large initial overclosure between the contacting boundaries.

If relaxation is used, enter the **Initial Relaxation Factor** r_i. The default is 0.005. This factor multiplies the penalty factor in the first outer iteration. Enter **Number of iterations with relaxation** N_r. The default value is 4. The penalty factor is gradually increased up to its full value, which is used in the outer iteration after the one where the specified number of iterations with relaxation have been reached.

If **Use relaxation** is set to **Conditional**, enter a **Suppression criterion**. The default is 0, which means that the relaxation is used for all parameter or time steps. This is a Boolean expression which, when fulfilled, suppresses the relaxation. If you, for

example, know that your problem needs relaxation only during the initial phase of the solution, enter an expression like `load_parameter>0.1`. It is also possible to use expressions based on the solution, for example `abs(solid.icnt1.gap)<0.05*h`. This expression is true when the gap or overclosure is small compared to the mesh size. It could be taken as an indication that the contact problem is almost converged, and thus not in need of any relaxation of the penalty factor.

Using the **Penalty factor control** selection **User defined** gives you the possibility to enter an explicit expression for penalty factor. Enter the **Contact pressure penalty factor** p_n. The default is `(min(1e-3*(5^niterCMP,1))*`*`<phys>`*`.<contact_tag>`*`.E_char)/`*`<phys>`*`.<contact_tag>`*`.hmin_dst`. The default value causes the penalty factor to be increased during the outer iterations and takes material stiffness and element size at the contact surface into account. The variable `E_char` contains the value given as characteristic stiffness, and `hmin` is the minimum element size on the selected boundaries.

The default values when **Manual tuning** is selected corresponds to the default expression when **User defined** is selected. The same expression is also used internally when **Preset** is used with the **Stability** option.

Settings for the Fully Coupled Augmented Lagrangian Method and the Nitsche Method

Select the type of **Penalty factor control** — **Automatic** (default), **Manual tuning**, or **User defined**.

- For **Manual Tuning**, enter a **Penalty factor multiplier** f_p. The default value is 1, which corresponds to the **Automatic** option.
- For **User defined**, enter a **Contact pressure penalty factor** p_n. The default value is *`<phys>`*`.`*`<contact_tag>`*`.E_char/`*`<phys>`*`.`*`<contact_tag>`*`.hmin`. The variable `E_char` contains the value given as characteristic stiffness, and `hmin` is the minimum element size on the selected boundaries.

In the *COMSOL Multiphysics Reference Manual:*

- Time Dependent and Stationary
- About the Time-Dependent Solver and About the Parametric Solver

CONTACT SURFACE OFFSET AND ADJUSTMENT

Enter a value or an expression for **Offset** d_{offset}. The offset is subtracted from the gap in the normal direction.

Use the offset properties to adjust initial clearances (negative values) or interference fits (positive values) without having to change the geometry. These properties are also useful for studying the effects of geometrical tolerance when the structure is still modeled using its nominal size.

Contact Surface Offset and Adjustment

INITIAL VALUE

This section is only available when **Formulation** is **Augmented Lagrangian**. In the augmented Lagrangian method, the contact pressure is introduced as an extra dependent variable on the destination boundaries. Enter a value for **Contact pressure** T_{n} to supply an initial value for the contact pressure.

- It is possible to speed up the convergence by supplying a guess of the correct order of magnitude of the contact pressure.
- If not all the contacting parts are fully constrained, it is sometimes possible to avoid singular problems by introducing a nonzero value for the contact pressure.

DISCRETIZATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

This section is only available when **Formulation** is **Augmented Lagrangian**. In the augmented Lagrangian method, the contact pressure is introduced as an extra dependent variable on the destination boundaries. You can modify the shape function type and order of this field. The default is to always use a linear shape function order for the contact pressure, but using a higher order can improve the accuracy of the contact condition. A value up to the discretization order of the displacement field is allowed. This setting also affects the shape function for dependent variables added by any subnode added to the **Contact** node.

To change the discretization, select **Shape function type** — **Lagrange** or **Nodal serendipity**.

- For the **Lagrange** shape-function type, select **Element order** — **Linear**, **Quadratic**, **Cubic**, **Quartic**, or (in 2D) **Quintic**.
- For the **Nodal serendipity** shape-function type, select **Element order** — **Quadratic**, **Cubic**, or **Quartic**. The **Nodal serendipity** option is not available when the order of the displacement field is linear.

The lumped solver used for the segregated augmented Lagrangian method is only optimal when the shape function order is **Linear**; otherwise, use a standard segregated step instead. The proper solver sequence is set up when adding a new default solver.

QUADRATURE SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Typically, the expressions in the contact weak equations include discontinuous functions. It is sometimes preferable to use a high integration order to improve the accuracy of the numerical integration of such expressions. By default, the numerical integration of the contact weak equations is twice the shape function order of the displacement field. Clear the **Use automatic quadrature settings** checkbox to specify a custom order. When the checkbox is cleared, enter an integer value between 0 and 41 in the **Integration order** field. This setting also affects the integration order for any subnode added to the **Contact** node.

- The **Integration order** affects the numerical integration of the contact weak contributions as well as the number of state variables used when, for example, computing friction forces.
- Increasing the integration order increases the time spent on assembling the system matrices each iteration.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Enter a **Characteristic stiffness** E_{char}. The default is *<phys>*`.Eequ`, the equivalent Young's modulus as defined by most material models. The characteristic stiffness (stored in a variable named *<phys>*`.`*<contact_tag>*`.E_char`) is used in expressions

for the default penalty factors for both the penalty and the augmented Lagrangian methods. The characteristic stiffness should be representative for the stiffness of the destination domain material in a direction normal to the boundary. You may for example need to adjust the default value in for the following cases:

- The material is strongly anisotropic. The default value is based on an average in all directions.
- The material is highly nonlinear. The default value is based on the stiffness at zero strain.
- The variable `<phys>.Eequ` is not defined by the material. This is the case for some user-defined materials.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Interior Contact** (Boundary)
Solid Mechanics, Explicit Dynamics > **More** > **Interior Contact** (Boundary)

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More** > **Interior Contact**

Friction

By adding the **Friction** subnode to a Contact or Interior Contact node, you can model static and dynamic friction in 3D, 2D, and 2D axisymmetry.

The selection in the **Friction** node is the same as that of its parent **Contact** or **Interior Contact** node. There can only be one **Friction** node under a specific **Contact** node.

In the case that the sliding velocity is known, you can replace the **Friction** node by Slip Velocity, which provides a more efficient model, since there is no need to solve for the friction forces and slip direction. The **Friction** and **Slip Velocity** nodes are mutually exclusive.

If **Friction** and Adhesion are present under the same **Contact** node, the friction settings will be ignored at the locations where the adhesion criterion is fulfilled.

COORDINATE SYSTEM SELECTION

This section is only present when **Friction** is used as a subnode to **Contact**. For **Interior Contact**, the coordinate system is selected in the parent node.

The friction forces are defined as boundary tractions with respect to the selected coordinate system. The selection is limited to boundary systems.

Make sure that the tangents of the selected boundary system are well defined on all destination boundaries.

FRICTION PARAMETERS

Select a **Friction model** — **Coulomb**, **Exponential dynamic Coulomb**, or **User defined**. The **Exponential dynamic Coulomb** option is only relevant for time-dependent studies, since the value of the friction coefficient depends on the slip velocity.

For **Coulomb** friction, enter a **Friction coefficient** μ.

For **Exponential dynamic Coulomb** friction, enter a **Static friction coefficient** μ_{stat}, a **Dynamic friction coefficient** μ_{dyn} and a **Friction decay coefficient** α_{dcf}.

The effective friction coefficient is

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}}) e^{-\alpha_{\text{dcf}} \|\mathbf{v}_{\text{slip}}\|}$$

where $\mathbf{v}_{\text{slip}}$ is the slip velocity.

For both Coulomb type options, you can modify the rules for sliding by providing minimum and maximum shear tractions.

Enter a **Cohesion sliding resistance** T_{cohe} to set a tangential traction that must be overcome before sliding can occur. The use of cohesion will give an offset to the friction force under sliding conditions.

Independently, you can enter a **Maximum tangential traction** $T_{\text{t,max}}$. When the tangential traction exceeds this value, slip will occur, independent of the normal pressure. The default expression is `Inf`, indicating that no limit on the tangential traction is active.

The tangential friction force during sliding, $T_{\text{t,crit}}$, can be written as

$$T_{\text{t, crit}} = \min(\mu T_{\text{n}} + T_{\text{cohe}}, T_{\text{t, max}})$$

Here, T_n is the contact pressure. If the tangential force is less than this value, there is no sliding.

When **Friction model** is set to **User defined**, enter an expression to define $T_{t,crit}$. This expression can depend on any quantity. However, the implementation of the friction model is only correct if $T_{t,crit}$ does not depend on the current friction force.

FRICTION FORCE PENALTY FACTOR

The settings in this section are used to control the penalty factor used by the friction model. The available settings depend on the chosen contact method in the parent **Contact** or **Interior Contact** node. When the **Nitsche** method is used in the parent **Contact** node, the same penalty factor as in the parent contact node is used for friction. Hence, this section is not available.

Settings for the Penalty and the Penalty, Dynamic Methods

When the **Penalty** or **Penalty, dynamic** method is used in the parent **Contact** or **Interior Contact** node, select the **Penalty factor control** —**Automatic** (default); **Automatic, soft**; **Manual Tuning**; **From Parent**; or **User defined**.

- **Automatic** and **Automatic, soft** provide predefined values for the penalty factor multiplier f_t.
- For **Manual Tuning**, enter a **Penalty factor multiplier** f_t. The default value is 1, which corresponds to the **Automatic** option.
- For **From Parent**, the friction force penalty factor $p_t = p_n/3$. This option is not available for the **Penalty, dynamic** method if the **Viscous only** penalty factor control is used in the parent **Contact** node.
- For **User defined**, enter a **Friction force penalty factor** p_t. The default value differs depending on the type of parent node:
  Contact: `(<phys>.<contact_tag>.E_char/<phys>.hmin_dst)/3`.
  Interior Contact: `(<phys>.<contact_tag>.E_char/<phys>.<contact_tag>.hmin)/3`.

Settings for the Augmented Lagrangian Methods

When the **Augmented Lagrangian** or **Augmented Lagrangian, dynamic** method is used in the parent **Contact** node, select the **Penalty factor control** — **From Parent**, **Preset**, **Automatic**, **Manual Tuning**, or **User defined**.

- **From Parent** is available when **Formulation** is **Augmented Lagrangian**, and is then the default. When selected, the friction force penalty factor $p_t = p_n/3$

- **Preset** is available when **Solution method** is **Segregated** in the parent **Contact** or **Interior Contact** node and for the **Augmented Lagrangian, dynamic**, for which is the default. It provides the same settings as described in the Contact Pressure Penalty Factor.
- **Automatic** is available when **Solution method** is **Fully coupled** in the parent **Contact** or **Interior Contact** node and provides a predefined value for the penalty factor multiplier f_t.
- For **Manual Tuning**, enter a **Penalty factor multiplier** f_t. The default value is 1. When **Solution method** is **Segregated** in the parent **Contact** or **Interior Contact** node and for the **Augmented Lagrangian, dynamic**, additional settings are available. These are the same as described for the Contact Pressure Penalty Factor section.
- For **User defined**, enter a **Friction force penalty factor** p_t. The default value depends on the selected **Solution method** in the parent **Contact** node.

INITIAL VALUE

If **Formulation** is **Augmented Lagrangian** or **Augmented Lagrangian, dynamic** in the parent **Contact** or **Interior Contact** node, enter values or expressions for the components of the initial force acting on the destination surface as **Friction force** $\mathbf{T}_t$.

To determine whether friction effects are active when starting the solution or not, select the **Previous contact state** — **Not in contact** or **In contact**.

The **Previous contact state** option is only available when **Friction** is used as a subnode to **Contact**.

For **In contact** enter values or expressions for the **Previous mapped source coordinates** $\mathbf{x}_{m,\,old}$. These serve as initial values to compute the tangential slip. The default value is (X, Y, Z) and indicates that the contacting boundaries are perfectly coincident in the initial state. The mapped source coordinates are defined as the location on the source boundary where it is hit by a certain point on the destination boundary.

The entries in this section should be given as vector components in the material frame.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

When the parent node is **Interior Contact**, the only setting in this section is **Compute frictional dissipation** .

For numerical reasons, the gap value will not be exactly zero even when the boundaries are in contact. A certain small positive value of the gap must thus be used to determine that the boundaries touch each other, so that friction forces can be introduced. Select a **Friction detection** — **Automatic** or **Manual**. For **Manual**, enter an absolute value for the **Friction detection tolerance** Δ_{friction}, which is the gap when the friction becomes active.

Select **Compute frictional dissipation** to compute and store to the energy dissipated by friction. This adds one extra dependent variable and an extra distributed ODE.

You can also request that the total accumulated slip distance it stored by selecting the **Store accumulated slip** checkbox.

The automatic friction detection tolerance is in the Solid Mechanics interface based on the mesh element size. For the Shell and Membrane interfaces, it is based on their thickness.

See also

- Including Friction, and The Friction Node in the *Structural Mechanics Modeling* chapter.
- Tangential Contact with Friction in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Friction**
Solid Mechanics > **Interior Contact** > **Friction**
Solid Mechanics, Explicit Dynamics > **Contact** > **Friction**
Solid Mechanics, Explicit Dynamics > **Interior Contact** > **Friction**
Shell > **Contact** > **Friction**

Layered Shell > **Contact** > **Friction**
Membrane > **Contact** > **Friction**
Multibody Dynamics > **Contact** > **Friction**

Ribbon
Physics tab with **Contact** or **Interior Contact** selected in the Model Builder tree:

Attributes > **Friction**

Slip Velocity

The **Slip Velocity** node facilitates a simplified form of slip friction modeling, which can be used in the case that the direction and speed of the sliding is known. In this case, there is no need to solve for the friction forces and slip orientation.

The selection in the **Slip Velocity** node is the same as that of its parent **Contact** node. There can only be one **Slip Velocity** node under a specific **Contact** node.

The **Slip Velocity** and Friction nodes are mutually exclusive.

The **Slip Velocity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

The slip velocity is defined as acting in the tangent plane of the destination boundary, and is given along the directions of a boundary system.

Only boundary systems can be selected. Make sure that the tangents of the selected boundary system are well defined on all destination boundaries.

FRICTION PARAMETERS

Select a **Friction model** — **Coulomb**, **Exponential dynamic Coulomb**, or **User defined**. Since the velocity is prescribed, a velocity dependent friction model can be used also in a stationary analysis.

For **Coulomb** friction, enter the **Friction coefficient** μ.

For **Exponential dynamic Coulomb** friction, enter a **Static friction coefficient** μ_{stat}, a **Dynamic friction coefficient** μ_{dyn} and a **Friction decay coefficient** α_{dcf}.

The effective friction coefficient is

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}}) e^{-\alpha_{\text{dcf}} \|\mathbf{v}_{\text{slip}}\|}$$

where $\mathbf{v}_{slip}$ is the slip velocity.

For both Coulomb type models, you can modify the rules for sliding by providing minimum and maximum shear tractions.

Enter a **Cohesion sliding resistance** T_{cohe} to provide an offset to the tangential traction.

Independently, you can enter a **Maximum tangential traction** $T_{t,max}$. This is the maximum tangential traction, independent of the contact pressure. The default expression is `Inf`, indicating that no limit on the tangential traction is active.

The magnitude of the tangential friction traction, $T_{t,crit}$, can be written as

$$T_{t,\,crit} = \min(\mu T_n + T_{cohe}, T_{t,\,max})$$

Here, T_n is the contact pressure.

When **Friction model** is set to **User defined**, enter an expression to define $T_{t,crit}$. This expression can depend on any quantity. However, the implementation of the friction model is only correct if $T_{t,crit}$ does not depend implicitly on the current friction force.

PRESCRIBED VELOCITY

Enter the relative velocity between the destination and source boundaries, $\mathbf{v}_{slip}$. The velocity is expressed in the coordinate system provided in the **Coordinate System Selection** section, interpreted on the destination boundary.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

For numerical reasons, the gap value will not be exactly zero even when the boundaries are in contact. A certain small positive value of the gap must thus be used to determine that the boundaries touch each other, so that friction forces can be introduced. Select a **Friction detection** — **Automatic** or **Manual**. For **Manual**, enter an absolute value for the **Friction detection tolerance** $\Delta_{friction}$, which is the gap when the friction becomes active.

Select **Compute frictional dissipation** to compute and store to the energy dissipated by friction. This adds one extra set of state variables to be stored. The integration is explicit, and adds only marginally to the computational cost.

You can also request that the total accumulated slip distance it stored by selecting the **Store accumulated slip** checkbox.

- The automatic friction detection tolerance is based on the mesh element size.
- The dissipated energy and accumulated slip require a time integration, and are thus only available for time-dependent analysis.
- The rate of energy dissipation is always computed, even in a stationary analysis. It can for example be used as a heat source in a heat transfer analysis. It is stored in a variable with a name like `solid.cnt1.qfric`.

See also

- Including Friction, and The Slip Velocity Node in the *Structural Mechanics Modeling* chapter.
- Tangential Contact with Friction in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Slip Velocity**
Shell > **Contact** > **Slip Velocity**
Solid Mechanics > **Contact** > **Slip Velocity**
Membrane > **Contact** > **Slip Velocity**
Multibody Dynamics > **Contact** > **Slip Velocity**

Ribbon

Physics tab with **Contact** selected in the Model Builder tree:

Attributes > **Slip Velocity**

Adhesion

Using the **Adhesion** subnode, you can add adhesion properties to a pair of boundaries in contact. This functionality requires that the contact is modeled using the penalty method, and that no offset is used in the penalty stiffness function.

The selection of the **Adhesion** node is the same as that of its parent Contact or Interior Contact node. There can only be one **Adhesion** node under a specific parent node.

The **Adhesion** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

This section is only present when **Adhesion** is used as a subnode to **Contact**. For **Interior Contact**, the coordinate system is selected in the parent node.

The adhesive stresses are defined as boundary tractions with respect to the selected coordinate system. The selection is limited to boundary systems. Make sure that the tangents of the selected boundary system are well defined on all destination boundaries.

ADHESIVE ACTIVATION

Select an **Activation criterion** to describe the onset of the adhesion between the source and destination boundaries.

- When the criterion is **Pressure**, enter the minimum contact pressure p_0, at which adhesion is initiated.
- When the criterion is **Gap**, enter the distance between the source and destination boundaries δ_0, at which adhesion is initiated. A negative value indicates that there must be a certain overclosure before adhesion is activated.
- When the criterion is **User Defined**, enter a Boolean expression such that adhesion is activated when it evaluates to true.

Enter a Boolean expression in the **Reset criterion** field to deactivate the adhesive force, such that adhesion is removed when it evaluates to true.

When the criterion is **Always active**, the source and destination boundaries are always in adhesive contact, unless broken by Decohesion.

ADHESIVE STIFFNESS

In this section, you specify the stiffness of the adhesive layer in the normal and tangential directions. The adhesive normal stiffness is used in tension only; in compression the contact pressure penalty factor is always used.

Select how to specify the **Adhesive stiffness** of the adhesive layer — **From contact penalty factor**, **User defined**, or **Use material data**. The default is to use **From contact penalty factor**, in which case the normal stiffness is the same as the **Contact pressure penalty factor** specified in the settings of the parent **Contact** node.

To define the tangential stiffness, select **Shear stiffness defined using** to be either **Normal to shear ratio** or **Adhesive Poisson's ratio**. For **Normal to shear ratio**, enter the ratio between the shear stiffness and the normal stiffness n_τ. For **Adhesive Poisson's ratio**, enter Poisson's ratio for the adhesive layer, ν, explicitly.

When **Adhesive stiffness** is set to **User defined**, enter each component of the stiffness vector $\mathbf{k}$ individually. The stiffness is given along the directions of the selected local coordinate system.

When **Adhesive stiffness** is set to **Use material data**, the stiffness vector $\mathbf{k}$ is computed from material data and layer thickness. From the **Specify** list, select a pair of elastic properties — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, or **Bulk modulus and shear modulus**. Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another according to Table 4-8. For the chosen properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

Enter a **Thickness**, d_s, to specify the physical thickness of the adhesive layer.

FRICTION INTERACTION

In this section you can control how adhesive and friction forces interact. In the **When adhesion activated** list, specify how friction forces are treated once the **Activation criterion** for adhesion is true. **Add friction forces** is the default option, which means that friction forces and adhesive force are additive. This option has the advantage that there will be a smooth transition from an adhesive bond to a pure friction state when for example a bond is broken by decohesion. Select **Ignore friction forces** to disregard any friction forces once adhesion is activated. There will be no residual strength due to friction even if the adhesive bond is broken.

See also

- Including Adhesion and Decohesion, and The Adhesion Node in the *Structural Mechanics Modeling* chapter.
- Adhesion in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Adhesion**
Solid Mechanics > **Interior Contact** > **Adhesion**
Solid Mechanics, Explicit Dynamics > **Contact** > **Adhesion**
Solid Mechanics, Explicit Dynamics > **Interior Contact** > **Adhesion**
Shell > **Contact** > **Adhesion**
Layered Shell > **Contact** > **Adhesion**
Membrane > **Contact** > **Adhesion**
Multibody Dynamics > **Contact** > **Adhesion**

Ribbon

Physics tab with **Contact** or **Interior Contact** selected in the Model Builder tree:

Attributes > **Adhesion**

Decohesion

Using the **Decohesion** subnode, you can add decohesion properties to a boundaries in contact. The behavior is slightly different, depending on whether a Contact or Interior Contact node acts as parent:

- When used with **Contact**, **Decohesion** is a direct child node. It requires that an **Adhesion** subnode is present and active under the same parent **Contact** node.
- When used with **Interior Contact**, **Decohesion** is added as a child to an **Adhesion** node.

The selection of the **Decohesion** node is the same as that of its parent **Contact** node. There can only be one **Decohesion** node under a specific **Contact** node.

The **Decohesion** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

This section is only present when **Adhesion** is used as a subnode to **Contact**. For **Interior Contact**, the coordinate system is selected in the parent node.

The adhesive stresses are defined as boundary tractions with respect to the selected coordinate system. The selection is limited to boundary systems. Make sure that the

tangents of the selected boundary system are well defined on all destination boundaries.

DECOHESION

Select a **Cohesive zone model** — **Displacement-based damage** or **Energy-based damage** to choose the type of variable that controls the damage process.

Select a **Traction–separation law** — **Linear**, **Exponential**, **Polynomial**, or **Multilinear**. The definition of these differ between the two cohesive zone models, and the last option is available only for **Displacement-based damage**.

For the displacement-based damage models, enter:

- **Tensile strength**, σ_t. This is the peak stress in pure tension.
- **Shear strength**, σ_s. This is the peak stress in pure shear.
- **Tensile energy release rate**, G_{ct}. This is the energy released during the whole decohesion process in a state of pure tension.
- **Shear energy release rate**, G_{cs}. This is the energy released during the whole decohesion process in a state of pure shear.

For the **Multilinear** separation law, also enter the **Shape factor**, λ.

When the traction–separation law is **Linear**, **Exponential**, or **Polynomial**, select the **Mixed mode criterion** — **Power law**, **Generalized power law**, or **Benzeggagh–Kenane**.

- For **Power law** or **Benzeggagh–Kenane** enter the **Mode mixity exponent** α.
- For **Generalized power law** enter the **Mode mixity exponents** α_t and α_s.

The mixed mode criterion determines how normal and shear components are combined into a single scalar failure criterion. For the **Multilinear** separation law, the mixed mode criterion is always linear (equivalent to a power law with $\alpha = 1$.)

For the energy-based damage models, enter:

- **Tensile damage threshold**, G_{0t}. This is the elastic energy at the onset of damage in pure tension.
- **Shear damage threshold**, G_{0s}. This is the elastic energy at the onset of damage in pure shear.
- **Tensile energy release rate**, G_{ct}. This is the energy released during the whole decohesion process in a state of pure tension.
- **Shear energy release rate**, G_{cs}. This is the energy released during the whole decohesion process in a state of pure shear.

- **Mode mixity exponent, damage initiation**, α_0. The value determines how normal and shear components are combined into a single scalar criterion for damage initiation.
- **Mode mixity exponent**, α_c. The value determines how normal and shear components are combined into a single scalar failure criterion.
- **Smoothing parameter**, N. This parameter adjusts the shape of the traction–separation law. It is available for the **Exponential** and **Polynomial** options. By default, $N = 1$, a smaller value gives a smoother behavior.

In the **Regularization** list, it is possible to add a viscous delay to the damage growth for time-dependent studies. Do this by selecting **Delayed damage** and enter a value for the **Characteristic time**, τ.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Maximum damage determines the residual stiffness of the adhesive layer after decohesion. By default, $d_{max} = 1$, which means that no residual stiffness remains. Enter a value smaller than 1 to introduce some residual stiffness.

Select **Compute damage dissipation energy** to compute and store to the energy dissipated by damage.

See also

- Including Adhesion and Decohesion, and The Decohesion Node in the *Structural Mechanics Modeling* chapter.
- Decohesion in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Decohesion**
Solid Mechanics > **Interior Contact** > **Decohesion**
Solid Mechanics, Explicit Dynamics > **Contact** > **Decohesion**
Solid Mechanics, Explicit Dynamics > **Interior Contact** > **Decohesion**
Shell > **Contact** > **Decohesion**
Layered Shell > **Contact** > **Decohesion**
Membrane > **Contact** > **Decohesion**
Multibody Dynamics > **Contact** > **Decohesion**

Ribbon

Physics tab with **Contact** or **Interior Contact** selected in the Model Builder tree:

Attributes > **Decohesion**

Wear

Add a **Wear** subnode to model mechanical wear created when the contacting surfaces are sliding along each other. Typically, the wear rate is a function of the contact pressure and the slip velocity.

The selection of the **Wear** node is the same as that of its parent **Contact** node. If significant wear can be expected on both surfaces in a contact pair, you can add two **Wear** nodes.

As wear involves a rate equation, it is only possible to compute wear in time-dependent studies. For other study types, the **Wear** node is ignored.

The **Wear** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

WEAR MODEL

Select a **Formulation** — **Deformed geometry** or **Offset-based**. When **Deformed geometry** is selected, the geometry of the domain is actually changed, and a set of extra degrees of freedom are added to the model. For the **Offset-based** formulation, the computed wear depth h_{wear} is added as an offset to the contact condition.

This setting is not present in the Shell and Membrane interfaces. In these interfaces, the wear is incorporated as a change in the thickness.

Select a **Wear model** — **Generalized Archard** or **User defined**.

For **Generalized Archard**, enter a **Wear constant**, k_{wear}, a **Reference contact pressure**, $T_{\text{n,ref}}$, and an **Exponent**, n. The wear rate is defined as

$$\frac{\partial h_{\text{wear}}}{\partial t} = k_{\text{wear}}\left(\frac{T_{\text{n}}}{T_{\text{n,ref}}}\right)^{n} \|\mathbf{v}_{\text{slip}}\|$$

Here, T_n is the contact pressure, and $\mathbf{v}_{slip}$ is the slip velocity. For a classical Archard equation, $n = 1$. The reference contact pressure, $T_{n,ref}$, can be chosen arbitrarily, and is used only to obtain consistent units. In most cases, $T_{n,ref}$, should be chosen as the unit pressure in the current unit system (for example, 1 Pa).

For **User defined**, enter an expression for the general source term, f, which defines the wear rate. The default is an expression similar to `(1e-9[m^2/N]*solid.cnt1.wear1.Tn)*solid.cnt1.wear1.vslipnorm`. This expression provides the names of the local variables for contact pressure and slip velocity, inserted in a classical Archard equation.

Finally, select a **Wear surface** — **Destination** or **Source**. If both surfaces in the contact pair are subject to wear, use two separate **Wear** nodes.

- In general, modeling wear on the destination side is slightly more accurate, since it is there that the contact pressure and slip velocity are originally computed.
- It is possible to use the Rigid Material material model on the source side.

DEFORMING DOMAIN SETTINGS

This section is only available when **Formulation** is set to **Deformed geometry**.

Choose how to set the **Deforming domain selection** — **Automatic** or **Manual**. When **Automatic** is selected, the deforming domains are assumed to be all domains adjacent to the destination or source boundaries, as selected by **Wear surface**. This is the default. By selecting **Manual**, a new section, **Deforming Domain Selection**, is displayed. There, you can make a manual selection of the domains to act as deforming.

In order for the mesh smoothing in the deforming domain to work properly, the mesh on all exterior boundaries, except the wear surface, should be free to slide in the tangential directions. Choose how to set the **Sliding boundary selection** — **Automatic** or **Manual**. When **Automatic** is selected, a sliding boundary condition added to all exterior boundaries of the deforming domain, expect the wear surface. This is the default. By selecting **Manual**, a new section, **Sliding Boundary Selection**, is displayed. There, you can make a manual selection of the sliding boundaries. The selection on which wear is applied is always excluded.

The remaining settings in the section affect the detailed mathematical model used for remapping the mesh in the deforming domain. For details, see Deforming Domain.

In most cases, the **Automatic** option will set up an appropriate **Deforming domain selection** and a corresponding **Sliding boundary selection**. For complex geometries it might, however, be necessary to manually control these selections. In case the automatic selections are incorrect, it is often displayed as an error from the solver, or clearly visible in the results of the exterior edges of the wear surface.

DISCRETIZATION

This section is present when **Formulation** is set to **Offset-based**, or when the **Wear** node is used in the Shell or Membrane interfaces.

In these cases, the amount of wear will be a separate field that is computed by a time integration of the wear rate expression. In the case of Solid Mechanics or Multibody Dynamics, the field is added to the contact offset. For Shell and Membrane, it is subtracted from the thickness of the elements.

Select a **Shape function type** — **From physics**, **Discontinuous Lagrange**, or **Gauss-point data**. When selecting **From physics**, the additional thickness field is approximated by the same shape functions that are used for the dependent variables of the physics.

The order of the shape functions is in all cases selected automatically, based on the discretization in the physics interface.

See also

- Including Wear, and The Wear Node in the *Structural Mechanics Modeling* chapter.
- Wear in the *Structural Mechanics Theory* chapter.
- Deformed Geometry and Moving Mesh in the *COMSOL Multiphysics Reference Manual.*

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Wear**

Solid Mechanics, Explicit Dynamics > **Contact** > **Wear**

Shell > **Contact** > **Wear**
Membrane > **Contact** > **Wear**
Multibody Dynamics > **Contact** > **Wear**

Ribbon
Physics tab with **Contact** selected in the Model Builder tree:

Attributes > **Wear**

Stabilization

It is common that parts in an assembly are initially unconstrained before contact forces are established in stationary and quasistatic simulations. Add a **Stabilization** subnode to suppress rigid body motion related to such unconstrained parts.

QUASISTATIC STABILIZATION

The stabilizing forces can be applied to different geometric entities as selected in **Apply to** — **Adjacent domains**, **Contact boundaries**, or **Contact pairs**. The available options depend on the physics interface.

For **Adjacent domains** and **Contact boundaries**, the stabilization works by connecting springs to ground for those geometric entities. When **Contact pairs** is selected, springs are instead added between the source and destination of the contact pairs of the parent **Contact** node.

COMSOL Multiphysics makes an estimation of the spring stiffness k_{char} required to stabilize the contact problem. If this estimate of the stiffness is too small or large, enter a new value for the **Stiffness multiplier** f_{stb}.

It is essential that the stabilization forces are relaxed once contact forces are established in order not to pollute the final results of the solution. Select a **Method** — **Automatic** or **Manual**, for how to relax the stabilization forces.

When the **Automatic** method is used, COMSOL Multiphysics identifies when the contact pair(s) first comes in contact. The stabilization forces will then remain for a sufficient number of steps and nonlinear iterations for contact forces to stabilize all adjacent parts after contact is established. The procedure is schematically described as

1. Detect the parameter or time step when contact is detected for at least one point on the destination
2. For the first step after contact, relax the stabilization stiffness to force a nonlinear correction

3 For the second step after contact, remove all stabilization forces after the first nonlinear iteration. Stabilization is kept for this iteration to ensure there are active friction forces to stabilize the previously disconnected parts

Since stabilization forces are relaxed over three steps, make sure the study step at least includes three parameter or times steps when using the **Automatic** method.

For the **Manual** method, select **Active for** — **First iteration**, **First step**, **Specified number of steps**, or **User-defined condition**, to control how the stabilization forces are relaxed.

- **First iteration** applies the stabilization forces for the first iteration of the nonlinear solver only. This can be useful if parts are initially in contact but require friction to be constrained.
- **First step** applies the stabilization forces for the first parameter step or time step. Enable **Iteration relaxation** to relax the spring stiffness with each iteration of the nonlinear solver.
- For **Specified number of steps**, enter the number of parameter steps or time steps for which the stabilization forces are active. By enabling the **Step relaxation**, the spring stiffness is reduced with each step number. If **Iteration relaxation** is enabled, the spring stiffness is relaxed with each nonlinear iteration of the final specified step number.

For **User-defined condition**, enter a Boolean expression. When evaluated to true, the stabilization forces are active. This option can be useful to control the activation based on a specific parameter value in a continuation sweep, or at a specific time in a time-dependent solver.

To monitor the stabilization energy added, add a **Global Variable Probe** of the variable `<phys>.<contact_feature>.Wstb`, and output probe values while solving in the **Fully Coupled** or **Segregated** solver. This setting allows you to track how stabilization relaxes with the nonlinear iterations.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Contact** > **Stabilization**
Solid Mechanics, Explicit Dynamics > **Contact** > **Stabilization**
Shell > **Contact** > **Stabilization**
Layered Shell > **Contact** > **Stabilization**
Membrane > **Contact** > **Stabilization**

Multibody Dynamics > **Contact** > **Stabilization**

Ribbon
Physics tab with **Contact** selected in the Model Builder tree:

Attributes > **Stabilization**

General Contact

Use the **General Contact** node to define mechanical contact between parts in an assembly as defined by a General Contact Pair.

- Mechanical contact contributions and properties are defined by adding one or several Contact Model subnodes.
- A No Contact subnode can be used remove contact contributions for selected boundaries.
- The geometrical properties of boundaries can be adjusted by adding an Offset subnode.

If general contact pairs are present and active in the model component, a default **General Contact** node is added to the physics interface with all pairs selected.

The pairs selected in a **General Contact** node are exclusive, which means that if a pair is selected more than once, only its last occurrence in the physics tree will be active.

The **General Contact** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

- In order to specify contact conditions, one or more **General Contact Pair** nodes must be available in the **Definitions** branch.
- A contact mapping always occurs from a source to a destination. For **General Contact**, the same boundary can simultaneously act as source and destination.
- All boundaries included in a **General Contact Pair** are referred to as source boundaries.
- Source boundaries that intersect with the selection of the **General Contact** node are referred to as destination boundaries.

When a **General Contact** node is present in the model, all study steps are geometrically nonlinear. The **Include geometric nonlinearity** checkbox in the study step **Settings** window is selected and cannot be cleared.

For the default **General Contact** node, selecting **Disconnect pair** removes the above requirement and makes it possible to clear the **Include geometric nonlinearity** checkbox.

When at least one **Contact**, **General Contact**, or **Interior Contact** node is active in the model, a **Contact Force** plot group is added to the list of result templates. The plot group contains one vector plot showing the contact pressure for each contact node. If a **Friction** node is present, then there will also be a vector plot of the tractions caused by friction.

- Contact Modeling
- Contact Analysis Theory

GENERAL CONTACT

Select the **Exclude constrained boundaries** to omit boundaries that are constrained from the selection of **General Contact**, effectively omitting any contact contributions from such boundaries. Here, constrained boundaries include both boundary constraints such as **Fixed Constraint**, **Prescribed Displacement**, or **Roller**, and boundaries that are adjacent to a domain constraint such as **Fixed Constraint** or **Prescribed Displacement**. By default constrained boundaries are excluded.

Excluding constrained boundaries removes them from the destination boundaries. However, they remain part of the source boundaries as defined by the general contact pair. A contact search from another boundary can thus still identify them as a valid contact mapping.

QUADRATURE SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Typically, the expressions in the contact weak equations include discontinuous functions. It is sometimes preferable to use a high integration order to improve the accuracy of the numerical integration of such expressions.

Select the **Integration order** — **From displacement field**; **From displacement field, reduced**; or **User defined**.

When selecting **From displacement field**, the numerical integration of the contact weak equations is twice the shape function order of the displacement field.

When selecting **From displacement field, reduced**, the numerical integration of the contact weak equations is the same shape function order of the displacement field.

When selecting **User defined**, specify a custom integration order, which can be an integer value between 0 and 41.

ADVANCED

The equations for the contact conditions can be implemented differently in order to influence the robustness and stability of the discretized system of equations and its linearization. This can be controlled by the **Jacobian Contribution** setting.

When the **Jacobian Contribution** is set to **Nonsymmetric**, the variations of the normal vector on the source boundary are excluded. This is the most robust implementation for cases where the source boundary undergoes large deformation. However, it has the drawback of causing the global stiffness matrix to become nonsymmetric, which increases the memory requirement and solution time in the linear solver.

The **Symmetric** option can be used instead when this is problematic for the linear solver. For this option, the main contact equations does not force a nonsymmetric stiffness matrix, although some subnodes may still do so. Variations of the normal vector on the source boundary are excluded also for this option, but it is not as robust as the **Nonsymmetric** option.

When the **Jacobian Contribution** is set to **Automatic**, the **Nonsymmetric** option is used for contact pairs using the **Deformed configuration** mapping method, while the **Symmetric** option is used for the **Initial configuration** mapping method.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Pairs** > **General Contact**
Solid Mechanics, Explicit Dynamics > **Pairs** > **General Contact**
Multibody Dynamics > **Pairs** > **General Contact**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Multibody Dynamics** selected:

Pairs > **General Contact**

Contact Model

The **Contact Model** node adds equations to implement the contact condition. It also defines the contact pressure used to enforce the normal contact constraint, such that boundaries in contact may not penetrate each other. The penalty method is always used to implement the contact condition for General Contact.

- Add a Damping (General Contact) subnode to add damping to the contact pressure.
- Add a Friction (General Contact) subnode to also include friction in the contact problem.

CONTACT MODEL

The penalty factor p_{n} defines the magnitude of the contact pressure for a given gap distance. It is defined as

$$p_{\mathrm{n}} = f_{\mathrm{p}}\, k$$

where f_{p} is a multiplier that can be used to tune the value of the penalty factor. The characteristic stiffness k is defined from a local penalty stiffness estimate k_{cnt}. For each destination boundary, k_{cnt} is defined from properties of the adjacent domain.

The definition of k_{cnt} is controlled by the **Penalty stiffness estimation** setting. Most common is to use **From material stiffness** for which k_{cnt} is defined from the equivalent stiffness E_{equ} of the adjacent domain material, if available, as

$$k_{\mathrm{cnt}} = \frac{E_{\mathrm{equ}}}{h}$$

where h is the characteristic size of the adjacent domain mesh element.

For transient study steps it is also possible to use **From mass**. Here, k_{cnt} is defined from the density ρ of the domain material and the current time step Δt of the solver:

$$k_{\mathrm{cnt}} = \frac{\rho h}{2\Delta t^2}$$

For other study steps, the stiffness estimate is always used.

Select how to define the **Penalty factor multiplier** — **Automatic**; **Automatic, soft**; or **Manual tuning**. For **Manual tuning** enter a dimensionless value for f_{p}.

There are additional controls for the characteristic penalty stiffness k used to define the penalty factor p_{n} in the **Advanced** section.

Select the type of **Penalty function** — **Ramp** or **Smooth ramp**. A smooth penalty function is often useful for impact problems to achieve a less abrupt transition between contact states.

Select **Offset penalty function** if you want the contact pressure to be nonzero when the gap is zero. For the **Ramp** function, enter a value for the **Contact pressure at zero gap** T_0. For the **Smooth ramp** function, enter the **Start location** Δ_{start} and the **Transition zone size** Δ_{size} of the function.

QUADRATURE SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Integration order** — **From parent**, **From displacement field**, **From displacement field, reduced**, or **User defined**.

When **From parent** is selected, the integration order is defined in the General Contact node. The other settings are described in Quadrature Settings.

ENERGY DISSIPATION

Select how to compute the energy dissipated in the **General Contact** node.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, or **Off**.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface; see, for instance, Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Off** to ignore the accumulation of dissipative processes.

ADVANCED

Select how to define the **Characteristic stiffness** k — **Harmonic mean**, **Minimum**, **Destination only** or **From fallback stiffness**.

- For **Harmonic mean**, k is defined as the harmonic mean of k_{cnt} evaluated at source and destination of the contact mapping.
- For **Minimum**, k is defined as the minimum of k_{cnt} evaluated at source and destination of the contact mapping.
- For **Destination only**, k is equal to k_{cnt} evaluated at the current, destination, boundary.
- For **From fallback stiffness**, k is set equal to the fallback stiffness k_{fb}.

Enter a value for the **Fallback stiffness** k_{fb}. The fallback stiffness is used if a definition of k_{cnt} is not available. For example, boundaries adjacent to a **Rigid Material** will always use k_{fb}.

The state of being in contact or not is used by other features or physics. Since the gap value for the penalty method will never be exactly zero, even when the boundaries are in contact, also a small positive the gap is typically considered as being in contact. The default **Multiphysics contact tolerance** is **Automatic**. If you want to explicitly specify the limit of the gap considered as being in contact, select **Manual**, and enter the **Contact tolerance** Δ_{cnt}.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **General Contact** > **Contact Model**
Solid Mechanics, Explicit Dynamics > **General Contact** > **Contact Model**
Multibody Dynamics > **General Contact** > **Contact Model**

Ribbon

Physics tab with **General Contact** selected in the Model Builder tree:

Attributes > **Contact Model**

No Contact

Use the **No Contact** node to define boundaries with not contact contribution. Excluding boundaries can be useful to reduce the computational cost if it can be determined before solving the model that some parts are unlikely to take part in contact problem.

The selection of the **No Contact** node is exclusive with Contact Model node.

The selected boundaries remain part of the source boundaries as defined by the general contact pair. A contact search from another boundary can thus still identify them as a valid contact mapping.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **General Contact** > **No Contact**
Solid Mechanics, Explicit Dynamics > **General Contact** > **No Contact**
Multibody Dynamics > **General Contact** > **No Contact**

Ribbon

Physics tab with **General Contact** selected in the Model Builder tree:

Attributes > **No Contact**

Offset

The **Offset** node can be used to modify the geometrical properties of its selected boundaries.

CONTACT SURFACE OFFSET AND ADJUSTMENT

Enter a value or an expression for **Offset from contact surface** d_{off}. The offset is subtracted from the geometric gap in the normal direction of the surface. Any specified offset will be considered on both source and destination of a contact mapping that includes the selected boundaries.

Select **Force zero initial gap** to compensate for any difference caused by irregularities in geometry or mesh when the two contacting boundaries should exactly coincide in the initial state. Gaps smaller than the tolerance Δ_{gap} are adjusted to be zero before any offset is added. By default, Δ_{gap} is set to `0`, meaning that all initial penetration are removed.

The adjustment by **Force zero initial gap** is only made if enabled at both source and destination of a contact mapping.

Use **Offset** to adjust initial clearances (negative values) or interference fits (positive values) without having to modify the geometry.

When combined with **Force zero initial gap**, the offset will be exact in the sense that it is not affected by mesh irregularities.

The adjustment made by **Force zero initial gap** does not move any nodes in the mesh. Effectively, it adds an extra hidden offset that compensates for the initial distance between the source and destination boundaries.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Change **Search distance** list from the default **From pair** to override the search distance set by a General Contact Pair. This is sometimes necessary for complex assemblies to avoid false or unwanted contact detection.

- If **Manual distance** is selected, enter an absolute value for the search distance
- If **Tolerance** is selected, enter a dimensionless tolerance. The search distance is equal to the entered tolerance times a characteristic mesh elements size.

The Offset Node in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **General Contact** > **Offset**
Solid Mechanics, Explicit Dynamics > **General Contact** > **Offset**
Multibody Dynamics > **General Contact** > **Offset**

Ribbon

Physics tab with **General Contact** selected in the Model Builder tree:

Attributes > **Offset**

Damping (General Contact)

By adding a **Damping** subnode to a Contact Model node, you can include viscous damping forces in the contact pressure definition for transient study steps. Including damping can, for example, be useful to stabilize the contact pressure and state for impact problems.

DAMPING

Select the **Damping type** — **Additive** or **Viscous only**.

When **Additive** is selected, a damping contribution is added to the elastic contact pressure defined by the parent **Contact Model** node.

Select **Viscous only** option to use a purely viscous contact pressure model. For both damping types, damping is non-zero only for negative gap velocities (that is, when the overclosure is increasing).

The viscous penalty factor p_{nv} is always defined proportional to the penalty stiffness factor p_n. Set **Specify** to **Characteristic time** to enter value for τ such that $p_{nv} = \tau p_n$. Alternatively, specify a **Damping ratio** ξ. In this case, the viscous penalty factor is defined by

$$p_{nv} = 2\xi\sqrt{p_n \frac{\rho h}{2}}$$

where ρ is the density of the adjacent domain material, and h is a characteristic size of the mesh element.

Select **Offset damping force** to shift the activation of damping. When **Offset damping force** is cleared, the damping force is added when the gap is zero. When selected, enter a value for the **Damping tolerance** Δ_{dmp}. A positive value means that the viscous force starts to act while there is still some gap.

The Damping Node in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **General Contact** > **Contact Model** > **Damping**

Solid Mechanics, Explicit Dynamics > **General Contact** > **Contact Model** > **Damping**

Multibody Dynamics > **General Contact** > **Contact Model** > **Damping**

Ribbon

Physics tab with **Contact Model** selected in the Model Builder tree:

Attributes > **Damping**

Friction (General Contact)

By adding the **Friction** subnode to a Contact Model node, you can include static and dynamic friction to the contact problem.

COORDINATE SYSTEM SELECTION

The friction forces are defined as boundary tractions with respect to the selected coordinate system. The selection is limited to boundary systems.

Make sure that the tangents of the selected boundary system are well defined on all destination boundaries.

FRICTION PARAMETERS

Select a **Friction model** — **Coulomb** or **Exponential dynamic Coulomb**. The **Exponential dynamic Coulomb** option is only relevant for time-dependent studies, since the value of the friction coefficient depends on the slip velocity.

For **Coulomb** friction, enter a **Friction coefficient** μ.

For **Exponential dynamic Coulomb** friction, enter a **Static friction coefficient** μ_{stat}, a **Dynamic friction coefficient** μ_{dyn} and a **Friction decay coefficient** α_{dcf}.

The effective friction coefficient is

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}}) e^{-\alpha_{\text{dcf}} \|\mathbf{v}_{\text{slip}}\|}$$

where $\mathbf{v}_{\text{slip}}$ is the slip velocity.

For both Coulomb type options, you can modify the rules for sliding by providing minimum and maximum shear tractions.

Enter a **Cohesion sliding resistance** T_{cohe} to set a tangential traction that must be overcome before sliding can occur. The use of cohesion will give an offset to the friction force under sliding conditions.

Independently, you can enter a **Maximum tangential traction** $T_{t,max}$. When the tangential traction exceeds this value, slip will occur, independent of the normal pressure. The default expression is `Inf`, indicating that no limit on the tangential traction is active.

The tangential friction force during sliding, $T_{t,crit}$, can be written as

$$T_{t,\,crit} = \min(\mu T_n + T_{cohe}, T_{t,\,max})$$

Here, T_n is the contact pressure. If the tangential force is less than this value, there is no sliding.

A general contact problem can involve the interaction between many different materials where each interaction might require different friction parameters. For such situations, the friction parameters can be made dependent on the mapping by using the `src2dst` operator combined with, for example, the `if` or `nif` operators.

The expression `if((src2dst_p1(dom)==2 && dom==3) || (src2dst_p1(dom)==3 && dom==2), 0.7, 0.3)` for the friction coefficient μ implies that $\mu = 0.7$ if the contact mapping occurs between boundaries 2 and 3, otherwise $\mu = 0.3$.

For an example of a conditionally defined friction parameters, see *Objects Falling in a Box*: Application Library path **Structural_Mechanics_Module/Contact_and_Friction/falling_objects**

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

For numerical reasons, the gap value is not exactly zero even when the boundaries are in contact. A small positive gap value must thus be used to determine that the boundaries are in contact, and to introduce friction forces. Select a **Friction detection tolerance** — **Automatic** or **Manual**. For **Manual**, enter an absolute value for the **Friction detection tolerance** $\Delta_{friction}$, which defines the gap distance when friction is activated.

You can also request that the total accumulated slip distance it stored by selecting the **Store accumulated slip** checkbox.

See also

- Including Friction and The Friction Node in the *Structural Mechanics Modeling* chapter.
- Tangential Contact with Friction in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **General Contact** > **Contact Model** > **Friction**
Solid Mechanics, Explicit Dynamics > **General Contact** > **Contact Model** > **Friction**
Multibody Dynamics > **General Contact** > **Contact Model** > **Friction**

Ribbon

Physics tab with **Contact Model** selected in the Model Builder tree:

Attributes > **Friction**

Continuity

Add a **Continuity** node to connect parts of an assembly using one or several **Identity boundary pairs**. Both sides of the identity pair must be part of the physics interface in which you add the **Continuity** node.

CONSTRAINTS SETTINGS

Select a method to implement the continuity conditions: **Pointwise constraints** (default), **Nitsche constraints**, or **Weak constraints**. For pointwise and weak constraints, the continuity conditions are implemented on the destination side of the pair.

Select the **Correct for mismatching boundaries** checkbox to apply a correction when the connecting boundaries overlap each other, or when there is a nonzero gap between them. By default, the checkbox is cleared.

The **Nitsche constraints** implement the continuity condition in a weak sense by adding equations instead of constraints. If many other nodes that add constraints are present in the model, this has the advantage of not adding extra work for the constraint elimination algorithm of the solver. Moreover, the Nitsche method is often the more

accurate method for connecting parts with nonconforming meshes at the shared boundary.

If **Nitsche** is selected, choose how to define the **Stabilization factor** — **Automatic, Manual Tuning**, or **User defined**. When **Manual tuning** is selected, enter a value for the **Stabilization factor multiplier**. This is a multiplier acting on the characteristic stiffness defined for all selected pairs. The stiffness is based on the material properties and the mesh element size on both sides of a pair. For **User defined**, enter an expression for the stabilization factor γ that will be used for all pairs.

Also, choose a **Formulation** — **Symmetric**, **Skew-symmetric** (default), or **Incomplete**. This setting controls how the equations added to implement the continuity condition are written and can influence the robustness and performance of the solution. The default **Skew-symmetric** formulation is the most robust formulation, but is more expensive to use compared to the **Incomplete** formulation. The **Symmetric** formulation has the advantage that it maintains the symmetry of the stiffness matrix, which can be important for large models.

It is also possible to select on which side of the pair to apply the continuity conditions by setting **Apply on** to either: **Source and destination** (default), **Source**, or **Destination**. It is generally recommended to apply the continuity conditions on the side with the finest mesh. However, the unbiased formulation obtained by the default **Source and destination** option is in the majority of cases equally accurate but can be more expensive.

ADVANCED

This section only shows for the default **Continuity** node. Disconnect the selected identity pairs by selecting **Disconnect pair**. When the pairs are disconnected, they will not add contributions by adding equations or variables, but the region is still identified as a pair region.

See also Identity, Contact, and Sector Symmetry Pairs and Default Pair Feature in the *COMSOL Multiphysics Reference Manual*.

	When using **Continuity** to connect parts of an assembly in a large deformation analysis, the most robust solution is obtained by setting the **Source fame** and **Destination frame** to **Material** in the used **Identity pairs**. This is especially important for the Nitsche method, since it enforces continuity in a weak sense only. For the same reason, avoid using a **Contact pair** in a **Continuity** feature also.

	The **Nitsche** method is not allowed for pairs, where any source or destination boundary is adjacent to a **Rigid Material**.

	See also Continuity Condition in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Pairs** > **Continuity**
Solid Mechanics, Explicit Dynamics > **Pairs** > **Continuity**
Multibody Dynamics > **Pairs** > **Continuity**

Ribbon

Physics tab with **Solid Mechanics** ; **Solid Mechanics, Explicit Dynamics** ; or **Multibody Dynamics** selected:

Pairs > **Continuity**

Bolt Pretension

Use the **Bolt Pretension** node to define the prestress force in pretensioned bolts. It is available for 3D and 2D axial symmetry only. Typically, the prestressed state is established in an initial study step, which is then followed by one or more steps in which the service loads are added.

When you add a **Bolt Pretension** node, a default Bolt Selection subnode is also automatically added. Use one such node to select each individual bolt.

The **Bolt Pretension** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

BOLT LABEL

Enter a name that will form the basis for the automatic generation of bolt labels when **Bolt Selection** subnodes are added. The label is used for identification during postprocessing. The default name is `Bolt`.

If the base name is changed, it only affects how bolt labels are generated in **Bolt Selection** subnodes that are added later on. You may then, in particular, want to manually edit the default **Bolt Selection** subnode.

BOLT PRETENSION

Select how to describe the pretension by selecting a **Pretension type** — **Pretension force**, **Pretension stress**, or **Tightening torque**.

- For **Pretension force** enter a value or an expression for F_p, the pretension force in the bolt.

  To compute the corresponding tightening torque, select the **Compute tightening torque** checkbox. Then, enter data as described below.

- For **Pretension stress** enter a value or an expression for σ_p, the pretension stress in the bolt. The pretension force is computed through multiplication by the actual area of each selected bolt.

  To compute the corresponding tightening torque, select the **Compute tightening torque** checkbox. Then, enter data as described below.

- For **Tightening torque**, enter the following data:
  - The tightening torque, M_T.
  - The **Nominal bolt diameter**. The default is **From geometry**, in which case the bolt diameters are derived from the cross section selections in the **Bolt Selection**

subnodes. To use another value, select **User defined**, and enter the bolt diameter, d.

- The **Bolt hole diameter**. The default is **Automatic**, in which case the bolt hole diameter is set to 1.1 times the nominal bolt diameter. To use anther value, select **User defined**, and enter the bolt hole diameter, d_{h}.
- The **Bolt thread mean diameter**. The default is **Automatic**, in which case the bolt thread mean diameter is set to 0.9 times the nominal bolt diameter. To use another value, select **User defined**, and enter the thread mean diameter, d_{t}.
- The **Bolt head type** — **Circular** or **Hexagonal**.

  For a circular bolt head, enter the **Bolt head diameter**. The default is **Automatic**, in which case the bolt head diameter is set to 1.6 times the nominal bolt diameter. To use another value, select **User defined**, and enter the bolt head diameter, d_{o}.

  For a hexagonal bolt head, enter the **Bolt head width**. The default is **Automatic**, in which case the bolt head width is set to 1.6 times the nominal bolt diameter. To use another value, select **User defined**, and enter the bolt head width, H.
- The **Half thread angle**, α. The default is `30[deg]`, which is used in for example the ISO and UTS standards.
- The **Lead**, l.
- The **Bolt head friction coefficient**, μ_{h}. This is the friction coefficient between the bolt head and the material under it, including the effect of, for example, washers. The default is 0.15.
- The **Bolt thread friction coefficient**, μ_{t}. This is the friction coefficient between inner and outer thread. The default is 0.15.

See also

- Modeling Pretensioned Bolts in the *Structural Mechanics Modeling* chapter for a complete description of how to model pretensioned bolts.
- Bolt Modeling Theory.
- Bolt Pretension Study for information about the **Bolt Pretension** study type.

Modeling of Pretensioned Bolts: Application Library path **Structural_Mechanics_Module/Tutorials/bolt_pretension_tutorial**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Bolt Pretension**

Solid Mechanics, Explicit Dynamics > **Bolt Pretension**

Beam > **Bolt Pretension**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; or **Beam** selected in the Model Builder tree:

Global > **Bolt Pretension**

Bolt Selection

The **Bolt Selection** subnode is automatically added as a default node to the Bolt Pretension node. It is used for selecting the individual bolts. One **Bolt Selection** node is required for each bolt.

BOUNDARY SELECTION

This section is only present in the Solid Mechanics interface.

From the **Selection** list, choose the boundaries to define a cross section of a single bolt. This cross section must be an interior boundary. It is the section where the stress in the bolt is measured.

In 2D axial symmetry, the cross section must be perpendicular to the *Z*-axis.

POINT SELECTION

This section is only present in the Beam interface.

From the **Selection** list, choose a point to define a cross section of a single bolt. This cross section must be an interior point. It is the section where the stress in the bolt is measured.

BOLT LABEL

Enter a **Bolt label**. The default is automatically generated based on the name given in the **Bolt label** section of the parent **Bolt Pretension** node. The label is used for identification during postprocessing.

The automatic name generation is also active when a **Bolt Selection** node is duplicated or copy-pasted.

BOLT PRETENSION

Select a **Pretension type** — **From Parent**, **Pretension force**, **Pretension stress**, or **Tightening torque**. In most cases, the input for the pretension force is given in the parent node. If, however, you need to modify the settings for individual bolts, you can do it here.

- For **Pretension force** enter a value or an expression for F_p, the pretension force in the bolt.
- For **Pretension stress** enter a value or an expression for σ_p, the pretension stress in the bolt. The pretension force is computed through multiplication by the actual area of each selected bolt.
- For **Tightening torque**, enter a value or an expression for M_T, the tightening torque. All other values needed for computing the bolt force from the torque are taken from the parent node.

If you want to include a **Relaxation** of the bolt pretension, enter a value or an expression for r. The relaxation is an axial displacement, which is subtracted from the predeformation of the bolt. This value can, for example, be time-dependent in the case of a creep relaxation of the bolt stress.

If you want to model a process where the bolt forces are not applied simultaneously, select the **Sequential tightening** checkbox. When selected, a new input field, **Pretensioning expression**, is accessible.

The **Pretensioning expression** is a Boolean expression, giving the parameter values at which the prestress is changed. Such an expression could, for example, be `par==5 || par==10` if the bolt is tightened in two steps. If the tightening process is done stepwise, you also need to use a **Pretension type** which is not the default **From parent**. The expression for the pretension force in each individual bolt can then vary with the continuation parameter, and should match the steps at which the pretension force is changed.

SYMMETRY DETECTION

If a bolt is cut by a symmetry plane, the force in the modeled part is only half of the force in the real bolt. When **Automatic symmetry detection** is selected, this is compensated for. The program will then automatically detect when a bolt is located in a symmetry plane, and apply corrections.

- If the input of the pretension load in the parent node is by **Pretension force**, the given force is interpreted as the force for the whole bolt.
- All results are given for the whole bolt.

When **Automatic symmetry detection** is cleared, the inputs and outputs are based on the modeled cross section area.

The **Symmetry Detection** section is not present in 2D axial symmetry, since the bolt in that case always is axially symmetric.

The **Symmetry Detection** section is not present in the Beam interface, since it is not possible to automatically determine the geometrical shape of the bolt.

See also Modeling Pretensioned Bolts in the *Structural Mechanics Modeling* chapter.

Modeling of Pretensioned Bolts: Application Library path **Structural_Mechanics_Module/Tutorials/bolt_pretension_tutorial**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Bolt Pretension** > **Bolt Selection**

Solid Mechanics, Explicit Dynamics > **Bolt Pretension** > **Bolt Selection**

Beam > **Bolt Pretension** > **Bolt Selection**

Ribbon

Physics tab with **Bolt Pretension** selected in the Model Builder tree:

Attributes > **Bolt Selection**

Bolt Thread Contact

Use the **Bolt Thread Contact** node to define a simplified model for the contact between two mating threads. The boundary between the internal and external thread is assumed to have a cylindrical shape. This feature is available for 3D and 2D axial symmetry only. One **Bolt Thread Contact** node is required for each bolt.

The **Bolt Thread Contact** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

PAIR SELECTION

From the **Pair selection** list, choose the contact pair that defines the thread connection for this bolt. The contact pair should be created so that the source boundary is the internal thread, and the destination boundary is the external (bolt) thread.

BOLT GEOMETRY

Enter the **Half thread angle**, α. The default angle is `30[deg]`, which is used in for example the ISO and UTS standards. For 2D axial symmetry, this is the only input required to describe the bolt geometry. For 3D, you need also to provide the following settings:

Enter the **Lead**, l. This is the axial distance a single thread travels after one full revolution of the bolt.

Select a **Handedness** — **Right-handed** or **Left-Handed**. Most bolts are right- handed.

Select a **Bolt geometry** — **Automatic** or **User defined**. If the mesh on the boundaries selected for the contact between internal and external thread give a good representation of a cylinder, the diameter and centerline of the bolt can be determined automatically. If this is not sufficient, you can instead enter the data manually.

- When using **Automatic**, the computed orientation of the beam axis, $\mathbf{e}_a$, is indeterminate with respect to its direction. If needed, you can control the direction by selecting the **Direction adjustment** checkbox. Then, enter a vector giving an **Approximate orientation**, $\mathbf{e}_{a,approx}$. The positive orientation of the bolt axis will be chosen so that $\mathbf{e}_a \cdot \mathbf{e}_{a,approx} > 0$.
- For **User defined**, enter the bolt diameter, d_p. Then enter the **Orientation** of the bolt axis, $\mathbf{e}_a$, and a **Point on axis**, $\mathbf{P}$.

Select a **Contact orientation** — **Both**, **Up**, or **Down**. In most cases, it is obvious that only one side of the thread will come into contact. It is then possible to decrease the size of the contact problem by considering only that direction. For a prestressed bolt, having

the bolt axis orientation vector $\mathbf{e}_a$ pointing toward the bolt head, it will usually be sufficient to check contact in the **Up** direction.

CONTACT

Enter a value or an expression for the **Penalty factor**, p. A smaller value will increase overclosure and sliding, but is in general more stable. The default value, `(2*solid.btc1.Eequ)/solid.btc1.rd`, is intended to supply a stiffness that gives a small elastic slip. This slip is related to the estimated bending stiffness of the threads.

Enter a value for the **Friction coefficient**, μ.

In order to specify contact conditions like this, one or more **Contact Pair** nodes must be defined in the **Definitions** branch of the component.

When a contact feature is present in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** checkbox on the study step **Settings** window is selected and cannot be cleared.

See also Simplified Modeling of Bolt Threads in the *Structural Mechanics Modeling* chapter for a complete description of how to model bolt threads.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Pairs** > **Bolt Thread Contact**

Solid Mechanics, Explicit Dynamics > **Pairs** > **Bolt Thread Contact**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected in the Model Builder tree:

Pairs > **Bolt Thread Contact**

Crack

Add a **Crack** node to indicate that certain boundaries represent a crack. A crack can either be infinitely thin, and represented by a single boundary, or being represented by disjoint surfaces in the geometry.

A crack can have any number of branches and corresponding crack fronts.

The **Crack** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available for 3D, 2D, and 2D axisymmetry.

CRACK FRONT

This section is only shown if **Crack surface** is set to **Symmetric** in the Crack Definition section. In this case, it is not possible to deduce the number of crack fronts uniquely from geometric analysis, so you must explicitly select the crack front or crack fronts.

	For 3D components, select a boundary or set of edges representing the crack fronts.
	For 2D and 2D axisymmetry, select points located at the crack fronts.

CRACK FACE 1

This section is only shown if **Crack surface** is set to **From geometry** in the Crack Definition section. In this case, the main selection will contain both sides of the crack. In this section, select one of the crack faces, so that it is possible to discriminate between the two faces of the crack.

COORDINATE SYSTEM SELECTION

The coordinate system selection is only used when a user defined crack orientation is selected in the Crack Orientation section.

CRACK DEFINITION

In this section, you select how the crack is represented in the geometry.

Select a **Crack surface** — **Slit**, **From geometry**, or **Symmetric**.

- When the crack is represented as an interior boundary, select **Slit**. This will cause the displacements of the two sides of the boundary to be decoupled from each other so that a crack is formed.
- When the crack is physically represented in the model, so that the two crack faces are different boundaries, select **From geometry**.
- If only half of a symmetric structure is modeled, so that the crack is an exterior boundary, select **Symmetric**. In 2D axisymmetry, this is only meaningful for cracks that are perpendicular to the axis of rotation. There are two cases:
  - The crack is infinitely thin and located in the symmetry plane. Other boundary conditions like **Fixed Constraint** or **Symmetry** will be overridden on the crack surface, so that the crack can be opened.
  - One of the crack faces is represented by the geometry. The other crack face is then assumed to be a mirror copy on the other side of the symmetry plane.

CRACK ORIENTATION

The direction of the crack growth is important if you are going to compute a J-Integral.

Select a **Crack orientation** — **From crack** or **User defined**. In most cases the crack growth direction can be automatically deduced from the geometry of the crack. If this is not the case, select **User defined** and then enter a vector for the **Crack growth direction**. The vector is represented in the coordinate system selected in the Coordinate System Selection section.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Crack** (Boundary)
Solid Mechanics, Explicit Dynamics > **More** > **Crack** (Boundary)

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected:

Boundaries > **More** > **Crack**

Crack Closure

Add a **Crack Closure** subnode under a Crack node to include contact conditions between crack faces. These includes contact in the normal direction of the crack faces, and, optionally, friction in the tangential directions. Only a single **Crack Closure** subnode can be added per **Crack** node.

The Crack Closure subnode is only applicable when **Crack surface** in the parent a Crack node is set to **Slit** or **Symmetry**. If **Crack surface** is set to **From geometry**, contact conditions can be added by a Contact node.

Friction can, furthermore, only be considered if **Crack surface** is set to **Slit**.

See also:

- Contact Between Crack Faces in the *Structural Mechanics Modeling* chapter.
- Contact Analysis Theory in the *Structural Mechanics Theory* chapter.

The **Crack Closure** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

CONTACT

Enter an expression for the **Penalty factor**, p_n. The default expression, `(2*<phys>.Eequ)/h`, estimates the penalty factor from the stiffness of the adjacent domains and the mesh element size. The constant can be modified to improve accuracy or stability of the model — a too high penalty factor can lead to ill-conditioned matrices.

Select a **Friction model** — **None**, **Coulomb**, or **User defined**. By selecting **None**, frictionless contact is considered.

When **Coulomb** is selected, the critical contact friction force is a function of the contact pressure. Enter values for:

- The **Friction coefficient**, μ,
- The **Cohesion sliding resistance**, T_{cohe},
- The **Maximum tangential traction**, $T_{\text{t,max}}$.

When **User defined** is selected, enter an expression for the **Critical friction force**, $T_{t,crit}$, that may depend on any quantity in the model. This expression defines the limit between sticking and sliding friction. Note, that the implementation of the friction model is only correct if $T_{t,crit}$ does not depend on the current friction force.

If either **Coulomb** or **User defined**, select **Compute frictional dissipation** to compute and store to the energy dissipated by friction.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Crack** > **Crack Closure**

Ribbon

Physics tab with **Crack** node selected in the model tree:

Attributes > **Crack Closure**

Face Load (Crack)

Add a **Face Load** subnode under a Crack node to describe a load acting on a crack face. In some cases, a Boundary Load can be used to the same effect. If, however, a J-integral is to be computed, and it encloses a loaded part of the crack face, then the load must be described using the **Face Load** node.

The **Face Load** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

LOAD

Enter the **Crack face pressure**, p. The pressure is applied to both faces of the crack.

See also J-Integral Theory in the *Structural Mechanics Theory* chapter.

You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Crack** > **Face Load**

Ribbon

Physics tab with **Crack** node selected in the model tree:

Attributes > **Face Load**

J-Integral

Add a **J-Integral** subnode under a Crack node in order to compute the J-integral along a certain integration path. When a J-integral has been computed, stress intensity factors will also be available.

An alternative to using J-integral evaluation is to use the Virtual Crack Extension method.

J-INTEGRAL

The J-integral is computed along a circular path centered at the crack front.

For 2D components, the integration path for the J-integral can be described either by a sequence of boundaries, or as a circular path around the crack tip. Select **Integration path** — **Circular** or **On edges**.

If **On edges** is selected, a new section, **Integration path** is shown. In that section, select the boundaries that define the integration path. The boundaries much be selected so that they form a contiguous path from one side of the crack surface to the other. In case the crack is located on a symmetry plane, the path should go from that plane to the crack surface.

For the case **Circular**, see below.

Circular Integration Path

Enter the **Radius of contour integral**, r_Γ. The default value is an expression of the form `solid.<crack_tag>.crackSize*0.5`. The variable `crackSize` is estimated from the crack geometry. For common cases, in particular in 2D, this is exactly the length of the crack. For more complex cases, in particular for branched cracks, there is no well-defined crack length, and the value of the variable must be considered only as being of the right order of magnitude.

The integration path must not intersect any exterior boundaries or internal slits, such as other cracks. There should also not be any holes or applied loads inside the integration path. Integration paths with a very small radius compared to the crack length will in general give low accuracy, since they will pass through the singular stress field in the vicinity of the crack tip. It is good practice to evaluate the J-integral along several paths, to assess the accuracy.

In 3D, the J-integral will vary along the crack front. It will be computed along several circular paths, centered at different locations along the crack front.

STRESS INTENSITY FACTORS

When computing stress intensity factors from the scalar-valued J-integral, an additional relation is needed in order to separate K_{I}, K_{II}, and (in 3D) K_{III} from each other. To perform this separation, the ratios between the displacement jumps in different directions across the crack surface are computed. The location for sampling the displacements should ideally be close enough to the crack tip, so that the field is governed by the crack tip singularity. On the other hand, it is not possible to evaluate too close to the crack tip, since the local solution is then not accurate as an effect of that same singularity. Rather than sampling displacements at a certain point, the ratios are averaged over a certain region. The default averaging region starts at 20% of the crack length, and ends at 50% of the crack length.

To change the size of the averaging region, enter the lower bound, b, and upper bound, c. If you are using a fine mesh around the crack tip, it is possible to use values significantly smaller than the default ones. Having the lower bound at a distance of two elements from the crack tip is sufficient.

For symmetric or antisymmetric load cases, where only one of the stress intensity factors is nonzero, the evaluation is not sensitive to the settings in this section.

INTEGRATION

In this section, you can adjust the accuracy of the numerical evaluation of the J-integral. The section is only shown when circular integration paths are used.

Integration on Contour

Select **Number of integration points** — **Fixed number** or **Relative to path length**.

When **Fixed number** is selected, enter the total number of points along the integration path, N_{Γ}.

When **Relative to path length** is selected, enter a relative number of points along the integration path, n_{Γ}. The total number of integration points is computed as

$$N_\Gamma = \frac{r_\Gamma}{L_{\text{crack}}} n_\Gamma$$

where r_Γ is the radius of the integration path, and L_{crack} is the crack length (as represented by the variable `solid.<crack_tag>.crackSize`).

Integration on Surface

In 3D and 2D axisymmetry, the J-integral also has contributions from the surface enclosed by the integration contour.

Select **Number of integration points** — **Fixed number** or **Relative to enclosed area**.

When **Fixed number** is selected, enter the total number of integration points on the enclosed surface, N_A.

When **Relative to enclosed area** is selected, enter a relative number of points on the enclosed surface, n_A. The total number of integration points is computed as

$$N_A = \left(\frac{r_\Gamma}{L_{\text{crack}}}\right)^2 n_A$$

where r_Γ is the radius of the integration path, and L_{crack} is the crack length (as represented by the variable `solid.<crack_tag>.crackSize`).

STRESS INTENSITY FACTORS

In order to separate the stress intensity factors for the different crack modes, the displacements on the crack surfaces are analyzed. In order to get a good value, this analysis should be performed in a region where the theoretical solution for the fields around a crack tip is valid. Ideally, this is close to the crack tip. It is, however, not possible to get an accurate numerical evaluation very close to the crack tip, due to the singularity of the fields.

The default is that the region used for the evaluation starts at a distance of 20% of the crack length from the tip, and extends to half the crack length. You can modify these values by entering a **Lower bound**, b, and an **Upper bound**, c.

See also

- J-Integrals in the *Structural Mechanics Modeling* chapter.
- J-Integral Theory in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus
Solid Mechanics > **Crack** > **J-Integral**

Ribbon
Physics tab with **Crack** node selected in the model tree:

Attributes > **J-Integral**

Reverse Crack Front

Use the **Reverse Crack Front** node to manually control the tangent direction of a crack front in 3D. This may be necessary, for example, if a crack front consists of several edges that have opposing tangent directions from the geometry. The main purpose of this node is to control the definition of what is positive and negative values of the stress intensity factor K_{II}.

This node has no other settings than an edge selection. The selected edges will, from the crack analysis point of view, have a tangent direction that is reversed when compared by the tangent direction defined by the geometry.

This node is only available in 3D.

The **Reverse Crack Front** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

LOCATION IN USER INTERFACE

Context Menus
Solid Mechanics > **Crack** > **Reverse Crack Front**

Ribbon
Physics tab with **Crack** node selected in the model tree:

Attributes > **Reverse Crack Front**

Virtual Crack Extension

Add a **Virtual Crack Extension** node to compute the energy release rate and stress intensity factors at a crack tip using the virtual crack extension method.

This is an alternative to using a J-Integral. The J-integral is easier to use, but the virtual crack extension method is more versatile.

The **Virtual Crack Extension** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

BOUNDARY SELECTION

Here you select the boundaries that constitute the crack surface to be used in the computation. The default is the same boundaries that are selected in the parent **Crack** node. In most cases, you do not have to change the selection.

CRACK FRONT

Select the edges (3D) or point (2D) that define the tip of the crack for which the energy release rate is to be computed.

SENSITIVITY

Enter the **Radius**, R_s, of the sensitivity region. This is the region that is deformed by the associated Deformed Geometry feature during the sensitivity analysis. The default radius is an expression of the form `solid.<crack_tag>.crackSize*0.5`. The variable `crackSize` is estimated from the crack geometry.

STRESS INTENSITY FACTORS

In order to separate the stress intensity factors for the different crack modes, the displacements on the crack surfaces are analyzed. In order to get a good value, this analysis should be performed in a region where the theoretical solution for the fields around a crack tip is valid. Ideally, this is close to the crack tip. It is, however, not possible to get an accurate numerical evaluation very close to the crack tip, due to the singularity of the fields.

The default is that the region used for the evaluation starts at a distance of 20% of the crack length from the tip, and extends to half the crack length. You can modify these values by entering a **Lower bound**, b, and an **Upper bound**, c.

ADVANCED

In this section, you can specify if the special virtual crack extension study should also contain a parametric sweep. Select **Add parametric sweep** — **Yes** or **No**.

If **Yes** is selected, then specify the **Sweep type** — **Specified Combinations** or **All combinations**. Then, add parameters, their ranges, and their units on the **Parameters**

table. The interpretation of the settings are the same as for the Parametric Sweep study step.

See also

- Virtual Crack Extension Method in the *Structural Mechanics Modeling* chapter.
- Virtual Crack Extension Theory in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Crack** > **Virtual Crack Extension**

Ribbon

Physics tab with **Crack** node selected in the model tree:

Attributes > **Virtual Crack Extension**

Enclosed Cavity

Add an **Enclosed Cavity** node to indicate that certain boundaries represent a closed volume. The **Enclosed Cavity** node defines variables for the cavity volume in the undeformed and deformed configurations.

The cavity can be filled with a compressible or an incompressible fluid by adding a **Fluid** subnode, see Fluid (Enclosed Cavity). In this case, the pressure of the fluid inside the cavity is governed by an equation of state, and it changes when the enclosed volume varies due to external loads. It is also possible to apply a constant pressure, or an initial overpressure using the **Prescribed Pressure** node, see Prescribed Pressure (Enclosed Cavity).

The **Enclosed Cavity** node can be used without any subnodes. In this case, variables for the enclosed volume in the undeformed and the deformed configurations are created, and they can be evaluated in the **Results** node.

The **Enclosed Cavity** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

An example can be found in *Hyperelastic Seal*: Application Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/hyperelastic_seal**

See also Enclosed Fluids in the *Structural Mechanics Theory* chapter.

REFERENCE POINT SELECTION

This section is only visible if the **Volume type** is **Open surface** and **Reference point** is **From selection**. In that case, select one point, representing the reference point $\mathbf{X}_{\text{ref}}$.

SKETCH

This section is only visible if the **Volume type** is **Open surface**. When the boundary selection forms an open surface, a reference point has to be carefully selected to correctly compute the enclosed volume. The sketch section provides hints for suitable reference points for different configurations.

VOLUME DEFINITION

When using the **Enclosed Cavity** node in the Membrane interface, select the **Enclosing surface** — **Top surface** or **Bottom surface** to indicate on which side of the membrane the cavity is located.

Select the **Volume type** — **Closed surface** (default) or **Open surface**. Choose **Open surface** when the selected boundaries do not form a closed volume due to, for instance, mirror and sector symmetries, or a plane end cap.

For **Open surface**, select the **Reference point** — **From selection** (default) or **User defined**. The reference point, $\mathbf{X}_{\text{ref}}$, must be carefully selected according to the rules defined above.

- When **From selection** is selected, select a reference point, $\mathbf{X}_{\text{ref}}$, in the **Reference Point** section.
- When **User defined** is selected, enter the coordinates for $\mathbf{X}_{\text{ref}}$.

Enter the **Volume scale factor** f_V.

See also Volume Definition and Symmetry Planes in the *Structural Mechanics Theory* chapter.

ADVANCED

Enter a value for **Fixed, external volume** V_{ext}, if applicable. If the fluid-enclosing volume can be divided into a deforming volume and a rigid volume (such as a rigid pipe or reservoir), the rigid volume does not need to be modeled explicitly. It will however affect the relative volume change, and thus, the pressure change acting on the elastic walls.

If certain enclosing boundaries are not part of the current physics selection (for instance, because they are considered rigid), they can still be marked as enclosing boundaries by selecting the **Include boundaries external to current physics** checkbox.

When selecting this checkbox, boundaries that are not part of the physics become applicable in the **Boundary Selection** section. For all exterior boundaries, a normal direction has to be specified. By convention, the normal direction must point into the cavity (or fluid domain). Define a suitable normal for all exterior boundaries by referencing a boundary system in **Normal pointing toward cavity defined by**. You can always create a new boundary system specific for this feature. If necessary, use the boundary system's subnode **Reverse Normal**, so that the normal for all exterior boundaries points toward the cavity.

At least one boundary has to be internal to the current physics, when including exterior boundaries in the **Boundary Selection** section.

In 2D, the out-of-plane thickness of exterior boundaries is assumed to be equal to the average thickness of selected physics boundaries.

The **Enclosed Cavity** node can be used to define a cavity across multiple physics interfaces when the total enclosed volume is defined by boundaries of two interfaces, for instance a Solid Mechanics and a Membrane interface. In this case, add the **Enclosed Cavity** node to only one of the two interfaces, and select the **Include boundaries external to current physics** checkbox to add the boundaries of the other physics in the **Boundary Selection** section. If the **Enclosed Cavity** node is added to, say, the Membrane interface, the pressure load must be manually included in the Solid Mechanics interface. The pressure variable has a name such as `<phys>.enc1.p_rel`, where `<phys>` is the name of the Membrane interface.

The volume change of due to exterior boundaries is only considered if the checkbox **Include geometric nonlinearity** is selected in the study step. For a geometrically linear study, exterior boundaries are considered rigid.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Enclosed Cavity** (Boundary)
Membrane > **More** > **Enclosed Cavity**

Ribbon

Physics tab with **Solid Mechanics** or **Membrane** selected:

Boundaries > **More** > **Enclosed Cavity**

Fluid (Enclosed Cavity)

Add a **Fluid** subnode under Enclosed Cavity to model the pressure load acting on the cavity walls due to changes in fluid volume. The fluid can be modeled as a compressible or incompressible fluid by defining an equation of state.

The **Fluid** subnode is contributing with the **Prescribed Pressure** subnode.

See also Enclosed Fluids in the *Structural Mechanics Theory* chapter.

FLUID PROPERTIES

Select the fluid **Compressibility** — **Compressible gas, adiabatic** (default), **Compressible gas, isothermal**, **Compressible fluid**, **Incompressible**, or **User defined**.

- When **Compressible gas, adiabatic** is selected, enter a **Reference pressure** p_{ref} (default 1 atm), the **Reference temperature** T_{ref} (default 293.15 K), and the **Ratio of specific heats** γ (default 1.4). The gas temperature is computed and stored in a variable called `<item>.T`, where `<item>` corresponds to the tag of the **Fluid** node, for example `solid.enc1.fl1`.
- When **Compressible gas, isothermal** is selected, enter the **Reference pressure** p_{ref} (default 1 atm).
- When **Compressible fluid** is selected, enter the **Reference pressure** p_{ref} (default 1 atm) and the **Compressibility of fluid** χ_{f} (default `4e-10` Pa^{-1}). By entering a compressibility which is much smaller than the surrounding solid (for example, `1/(1e3*K)` where `K` a representative bulk modulus of the solid), the cavity will become nearly incompressible. Extremely low compressibility may lead to numerical instabilities.
- When **User defined** is selected, enter an expression for the **Pressure** p. The default is an expression similar to `(-1[GPa])*solid.enc1.DeltaV/solid.enc1.V_ref`. The pressure is typically a function of the volume change. To apply a constant pressure load, use the **Prescribed Pressure** node instead.

For **Compressible gas, adiabatic**, **Compressible gas, isothermal**, and **Compressible fluid**, the reference pressure can be specified in two ways.

Specify the **Reference pressure** — **From prescribed pressure** or **Used defined**.

- When **From prescribed pressure** is selected, the reference pressure is equal to the pressure given in one or more **Prescribed Pressure** nodes. See Prescribed Pressure (Enclosed Cavity).
- When **User defined** is selected, enter an expression for p_{ref}.

ADVANCED

From the **Reference volume** list, select **Automatic**, **From undeformed configuration**, **From deformed configuration**, or **User defined**.

- When **Automatic** is selected, the reference volume is equal to the volume during the initialization of the study step (spatial frame configuration). In particular, this is useful when the cavity is already deformed due to an initial overpressure applied in a previous study step.

- When **From undeformed configuration** is selected, the reference volume is equal to the volume of the undeformed configuration (material frame configuration).
- When **From deformed configuration** is selected, the reference volume is equal to the volume during the initialization of the study step (spatial frame configuration). In particular, this is useful when the cavity is already deformed due to an initial overpressure applied in a previous study step.
- When **User defined** is selected, enter an expression for the reference volume, V_{ref}. The default is an expression similar to `withsol('sol1',solid.enc1.V)`.

From the **Formulation** list, select **Automatic**, **Pressure formulation**, or **Strain formulation**.

- When **Automatic** is selected, the method to compute the fluid pressure is automatically selected. This setting should be appropriate for most cases.
- When **Pressure formulation** is selected, the fluid pressure is computed with a global auxiliary DOF, which represents the fluid pressure.
- When **Strain formulation** is selected, the fluid pressure is computed with a global auxiliary variable that represents the volume change.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Enclosed Cavity** > **Fluid**
Membrane > **More** > **Enclosed Cavity** > **Fluid**

Ribbon

Physics tab with **Enclosed Cavity** selected in the model tree:

Attributes > **Fluid**

Prescribed Pressure (Enclosed Cavity)

Add a **Prescribed Pressure** subnode under Enclosed Cavity to apply a known pressure. This feature can be used to model a constant pressure inside the cavity, or when there is an initial overpressure in the cavity in addition to pressure changes resulting from wall deformations.

An example for the latter case is a vehicle tire, which in a first step is pressurized to a certain known overpressure. In a second step, the tire may be deformed due to external loads such as when the tire is pushed on the ground, and deforms further due to the vehicle's weight.

This type of analysis requires two study steps. In the first step, only the initial overpressure is applied to compute the deformed volume. In the second step, the initial overpressure must be maintained, while additional pressure changes occur due to the external deformation of the cavity. These additional pressure changes can be modeled by adding a **Fluid** subnode, see Fluid (Enclosed Cavity), with appropriate values for the reference pressure and reference volume. The reference volume must be equal to the deformed volume from the first step.

The **Prescribed Pressure** node is contributing with the **Fluid** node and other **Prescribed Pressure** nodes.

PRESCRIBED PRESSURE

Enter the **Pressure** p.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Enclosed Cavity** > **Prescribed Pressure**
Membrane > **More** > **Enclosed Cavity** > **Prescribed Pressure**

Ribbon

Physics tab with **Enclosed Cavity** selected in the model tree:

Attributes > **Prescribed Pressure**

Filter (Enclosed Cavity)

Add a **Filter** subnode under Enclosed Cavity to indicate that a certain part of a boundary does not enclose the fluid. A boundary region excluded by a filter selection is ignored in the volume computation, and the pressure load is not applied on such region.

Applying a filter to a boundary may be necessary if the geometry is built as an assembly. An example is a piston–cylinder assembly, in which the piston moves through the cylinder as illustrated in Figure 4-1. In the initial position as well as during the piston

movement, part of the cylinder boundary may not (or no longer) enclose the fluid. These boundary regions can be excluded using a filter.

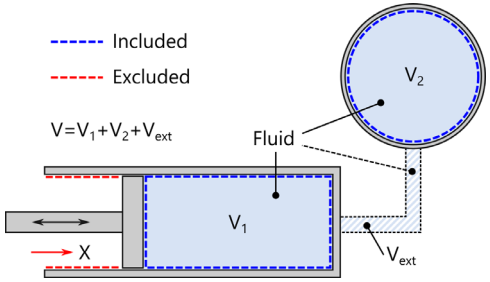


Figure 4-1: A piston–cylinder assembly which requires a filter.

Only boundaries selected in the parent Enclosed Cavity node can be selected in the **Filter** subnode. Select only boundaries on which a filter expression should be applied. Boundaries which have no filter condition are always included.

SELECTION FILTER

Enter a Boolean expression for the **Logical expression for inclusion, undeformed configuration** $e_{f,ref}$. The expression should evaluate to 1 (include) or 0 (exclude). The filter expression affects the computation of the reference volume, and it must not depend of the deformation.

Enter a Boolean expression for the **Logical expression for inclusion, deformed configuration** e_f. The expression should evaluate to 1 (include) or 0 (exclude). The filter expression affects the computation of the volume in the deformed configuration. No pressure forces from the Fluid (Enclosed Cavity) node or Prescribed Pressure (Enclosed Cavity) node are applied on boundary regions, where the Boolean expression evaluates to 0.

Example

Consider the piston–cylinder assembly shown in Figure 4-1, where the piston moves in the X direction. In this case, a filter expression must be applied on the cylinder wall. To track the piston position, it is convenient to define an average operator (material frame) on a suitable reference point, edge, or boundary on the piston's head.

Assuming that the operator is called `aveop1`, the Boolean expression for the **Logical expression for inclusion, undeformed configuration** $e_{f,ref}$ reads `X>=aveop1(X)`, and the **Logical expression for inclusion, deformed configuration** e_f reads `X>=aveop1(X+u)`, where `u` is the piston displacement in the X direction.

In a geometric nonlinear study, the expression `X+u` is equivalent to the spatial coordinate `x`.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Enclosed Cavity** > **Filter**
Membrane > **More** > **Enclosed Cavity** > **Filter**

Ribbon

Physics tab with **Enclosed Cavity** selected in the model tree:

Attributes > **Filter**

Port

The **Port** boundary condition is used to excite and absorb elastic waves that enter or leave waveguide structures. A given port condition supports one specific propagating mode. To provide the full description, combine several port conditions on the same boundary, for example, one for longitudinal, one for torsional, and one for transverse modes. Make sure that all relevant propagating modes in the studied frequency range are included (all modes that have a cutoff frequency in the frequency range). By doing this, the combined port conditions provide a superior nonreflecting condition for waveguides to, for example, a perfectly matched layer (PML) configuration or the Low-Reflecting Boundary. The same port boundary condition feature should not be applied to several waveguide inlets and outlets. The port condition supports S-parameter (scattering parameter) calculation, but it can also be used as a source to just excite a system.

The **Port** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

The **Port** boundary condition exists only for 3D models.

See the tutorial *Mechanical Multiport System: Elastic Wave Propagation in a Small Aluminum Plate*. Application Library path: **Structural_Mechanics_Module/Elastic_Waves/mechanical_multiport_system**

On a given boundary, a combination of ports will define the total displacement field $\mathbf{u}$ (sum of incident and outgoing waves) as

$$\mathbf{u} = \sum_{i \in \text{bnd}} (S_{ij}\mathbf{u}_i + A^{\text{in}} e^{i\varphi} \delta_{ij}(\mathbf{u}_i - 2\mathbf{n}(\mathbf{n} \cdot \mathbf{u}_i)))$$

where the summation "i" is over all ports on the given boundary "bnd", S_{ij} is the scattering parameter, A^{in} is the amplitude of the incident field (at port "j"), φ is a possible phase, and $\mathbf{u}_i$ is the mode shape of the ith port. Note that the mode shape $\mathbf{u}_i$ is normalized to carry unit apparent power, that is 1 W. This definition means that the scattering parameter S_{ij} defines the amplitude of mode i when a system is exited at port j (with mode j). This corresponds to a multimode expansion of the solution on the given port. The scattering parameters are automatically calculated when a model is set up with just one port exciting the system. To get the full scattering matrix, The Port Sweep Functionality can be used.

Only one port should be excited at a time if the purpose is to compute S-parameters. The S-parameters are defined as `solid.Smatrix11`, `solid.Smatrix21`, and so on, and can be used in postprocessing.

In other cases, having more than one port exciting the system might be wanted, but the S-parameter variables cannot be correctly computed. When several ports are excited, the S-parameter output is turned off.

When working with ports in the Solid Mechanics interface, it is important to remember that computed duct modes do not always represent an orthogonal basis set. This fact is taken into account in the formulation of the port equations, by including so-called cross-terms between the modes.

PORT PROPERTIES

Enter a unique **Port name**. Only nonnegative integer numbers can be used as **Port name** as it is used to define the elements of the S-parameter matrix. The numeric port names are also required for the port sweep functionality. The port name is automatically incremented by one every time a port condition is added.

Select a **Type of port**: **Numeric** or **User defined**. Depending on the selection, different options appear in the **Port Mode Settings** section (see below).

PORT MODE SETTINGS

Depending on the option selected in the **Type of port** (see above):

- In the case of a **Numeric** port, a boundary mode problem is solved on the port face to compute the desired propagating mode. This option requires the use of a **Boundary Mode Analysis** step in the study. It should be placed before the **Frequency Domain** step. In the study, add one **Boundary Mode Analysis** step for each **Numeric** port and make sure to reference the proper **Port name** in the study step. When using the **Numeric** port, it is not possible to perform a frequency sweep in the **Frequency Domain** study step. Only one frequency can be used and it should correspond to the **Mode analysis frequency** entered in the **Boundary Mode Analysis** step(s). One option is to add a **Parametric Sweep** and define a parameter for the frequency used in both the steps. In this case, care should be taken when setting up the search criteria in the mode analysis.
- For **User defined**, enter user defined expressions for the **Mode shape** $\mathbf{u}_\mathrm{n}$ and the **Mode wave number** k_n. The mode shape will automatically be scaled to unit power before it is used in the port condition. Use the user defined option to enter a known analytical expression.

When the **Numeric** port option is used and the boundary mode analysis is run, the boundary conditions from the Solid Mechanics model are automatically inherited in the boundary problem. For this automatic procedure, there is only support for the **Free**, **Fixed**, **Roller**, **Symmetry**, **Antisymmetry**, and **Spring Foundation** conditions.

INCIDENT MODE SETTINGS

Activate if the given port is excited by an incident wave of the given mode shape. For the first **Port** condition added in a model, the **Incident wave excitation at this port** is set to **On**. For subsequent conditions added the excitation is set to **Off** per default. If more than one port in a model is excited, the S-parameter calculation is not performed.

When the **Incident wave excitation at this port** is set to **On**, then select how to define the incident wave. Select the **Define incident wave**: **Displacement amplitude** or **Power**.

- For **Displacement amplitude** enter the amplitude A_u^{in} of the incident wave. This is in general defined as the maximum amplitude for a given mode shape.

- For **Power** enter the power P^{in} of the incident wave. This is in general defined as the RMS power of the incident wave.
- For both options enter the phase φ of the incident wave. This phase contribution is multiplied with the amplitude defined by the above options. The **Displacement amplitude** input can be a complex number.

Note that when the **Activate port sweep** option is selected at the physics level, the options in the **Incident Mode Settings** section are deactivated. This is because this option automatically sends in a mode of unit power, sweeping through one port at the time.

All ports with incident wave excitation turned off have an arrow symbol, added in the Graphics window, that points outward. When incident wave excitation is turned on, the arrow symbol points inward.

If a waveguide is cut by one or several symmetry planes, care should be taken when setting up the port modes and when postprocessing.

When postprocessing, remember that absolute values like, for example, the outgoing power at port 1, `solid.port1.P_out`, needs to be multiplied with an appropriate factor. Multiply with 2 if one symmetry plane is used, and so on.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options**.

THE PORT SWEEP FUNCTIONALITY

The port sweep functionality is used to reconstruct the full scattering matrix S_{ij} by automatically sweeping the port excitation through all the ports included in the model. When the port sweep is activated, the options in the **Incident Mode Settings** in the port conditions are deactivated and COMSOL controls which port that is excited with an incident mode.

The port sweep functionality is activated at the main physics interface level by selecting **Activate port sweep** in the Port Sweep Settings section. Enter the **Sweep parameter name**, the default is `PortName`. Create a parameter with the same name under **Global Definitions** > **Parameters 1**. This is the name of the parameter to be used in a parametric sweep, here it should represent the **Port name** integer values (defined when adding the

port conditions). Add a parametric sweep study step and run the sweep over the `PortName` parameter with an integer number of values representing all the ports in the model. Once the model is solved the full scattering matrix can be evaluated using the defined global variables `solid.Smatrix11`, `solid.Smatrix21`, `solid.Smatrix12`, ans so on.

Use the **Global Matrix Evaluation** under **Derived Values** to evaluate the full scattering matrix `solid.Smatrix`.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Port** (Boundary)

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries > **More** > **Port**

Elastic Predeformation

The **Elastic Predeformation** node can be used for incremental solution updates using the deformed geometry functionality. This approach can significantly speed up computations is case of large elastic deformations.

The equations solved by the Solid Mechanics interface are formulated using the material frame. The concept of deformed geometry implies that the geometry of the structure on the material frame can differ from that originally drawn (using the geometry frame).

The **Elastic Predeformation** subnode is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available for 3D, 2D, and 2D axisymmetry.

ELASTIC PREDEFORMATION

By default, the **Predeformation gradient** is set to **From material frame**, which means that the deformation gradient relating the geometry and material frames will be used. The predeformation gradient will enter into the definition of the elastic strains, which will create an initial stress state on the material frame corresponding to the total deformation starting from the original geometry configuration.

When the checkbox **Store deformation history** is checked (default) the displacement field computed during the current solution step (using the current material frame) will be stored in a special variable, `u_pd`, which can be used then on the next solution step, if any. When the checkbox **Set up deformed geometry** is selected (default) the material frame will differ from the geometry frame by a prescribed displacement `u_pd` computed at the previous solution step. The solution computed on each step represents an incremental step in the total deformation. The total solution can be set up using either a time-dependent or a stationary parametric study, for example, with the loading gradually increasing from one step to the other. Special postprocessing variables, `u_tot_old` and `u_tot`, are defined to store the total displacement before and after each step, respectively.

Alternatively, you can set the **Predeformation gradient** to **User defined**. Then, you can enter explicitly expressions referring to some previously computed solution, for example: `withsol('sol1', solid.FdxX)`, and so on. The contribution will enter the elastic strain definitions, which will result into an initial stress state corresponding to the previously computed solution.

- Adding this feature will induce the geometric nonlinearity in the model; the corresponding checkbox on every study step node will be automatically checked and become inaccessible (grayed out).
- This feature takes inelastic strain contributions into account. It is thus possible to use it together with nonlinear material models like large deformation plasticity.
- The domain selection is set to all domains and locked, so that all domains selected by the corresponding Solid Mechanics interface are always selected. This is because the material frame handling requires consistency over the entire structure.
- All user inputs for distributed loads in Solid Mechanics should be interpreted as being force densities in the original undeformed configuration (Geometry Frame) when an **Elastic Predeformation** feature is present.
- When this feature is present, the displacement input in the **Prescribed Displacement** nodes (domain, boundary, edge, or point) will represent a constraint for the total displacement.

In the *Structural Mechanics Theory* chapter:

- Deformed Geometry
- Elastic Predeformation

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **More** > **Elastic Predeformation**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains > **More** > **Elastic Predeformation**

Phase

You can add a **Phase** subnode to nodes that define a load in order to prescribe the phase angle in a frequency-domain analysis.

For modeling the frequency response, the physics interface splits the harmonic load into two parameters:

- The amplitude, F, which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f:

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi f t + \phi)$$

SHELL PROPERTIES

This section is only present in the Layered Shell interface. See also The Shell Properties and Interface Selection Sections and the documentation for the corresponding parent nodes.

PHASE

Enter the components of **Load phase** ϕ in radians (for a pressure the load phase ϕ is a scalar value). Add `[deg]` to a phase value to specify it using degrees.

Typically, the load magnitude is a real scalar value. If the load specified in the parent feature contains a phase (using a complex-valued expression), the software adds the phase from the **Phase** node to the phase already included in the load.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Body Load** > **Phase**
Solid Mechanics > **Boundary Load** > **Phase**
Solid Mechanics > **Edge Load** > **Phase**
Solid Mechanics > **Point Load** > **Phase**
Solid Mechanics > **Ring Load** > **Phase**
Solid Mechanics > **Point Load (on Axis)** > **Phase**
Solid Mechanics > **Rigid Connector** > **Applied Force** > **Phase**
Solid Mechanics > **Rigid Connector** > **Applied Moment** > **Phase**
Solid Mechanics > **Rigid Material** > **Applied Force** > **Phase**
Solid Mechanics > **Rigid Material** > **Applied Moment** > **Phase**
Solid Mechanics, Explicit Dynamics > **Body Load** > **Phase**
Solid Mechanics, Explicit Dynamics > **Boundary Load** > **Phase**
Solid Mechanics, Explicit Dynamics > **Edge Load** > **Phase**
Solid Mechanics, Explicit Dynamics > **Point Load** > **Phase**
Solid Mechanics, Explicit Dynamics > **Ring Load** > **Phase**
Solid Mechanics, Explicit Dynamics > **Point Load (on Axis)** > **Phase**
Layered Shell > **Body Load** > **Phase**
Layered Shell > **Boundary Load** > **Phase**
Layered Shell > **Edge Load** > **Phase**
Layered Shell > **Line Load** > **Phase**
Layered Shell > **Point Load** > **Phase**
Layered Shell > **Rigid Connector** > **Applied Force** > **Phase**
Layered Shell > **Rigid Connector** > **Applied Moment** > **Phase**
Layered Shell > **Rigid Material** > **Applied Force** > **Phase**
Layered Shell > **Rigid Material** > **Applied Moment** > **Phase**
Membrane > **Body Load** > **Phase**

Membrane > **Face Load** > **Phase**
Membrane > **Edge Load** > **Phase**
Truss > **Edge Load** > **Phase**
Truss > **Point Load** > **Phase**
Truss, Explicit Dynamics > **Edge Load** > **Phase**
Truss, Explicit Dynamics > **Point Load** > **Phase**
Wire > **Edge Load** > **Phase**
Wire > **Point Load** > **Phase**
Membrane > **Point Load** > **Phase**
Multibody Dynamics > **Body Load** > **Phase**
Multibody Dynamics > **Boundary Load** > **Phase**
Multibody Dynamics > **Edge Load** > **Phase**
Multibody Dynamics > **Point Load** > **Phase**
Multibody Dynamics > **Rigid Connector** > **Applied Force** > **Phase**
Multibody Dynamics > **Rigid Connector** > **Applied Moment** > **Phase**
Multibody Dynamics > **Rigid Material** > **Applied Force** > **Phase**
Multibody Dynamics > **Rigid Material** > **Applied Moment** > **Phase**

Ribbon
Physics tab with **Body Load**, **Boundary Load**, **Face Load**, **Edge Load**, **Point Load**, **Applied Force**, or **Applied Moment** selected:

Attributes > **Phase**

Harmonic Perturbation

Use the **Harmonic Perturbation** subnode to specify the harmonic part of nonzero prescribed displacements, rotations, velocities, or accelerations. This node is used if the study step contains frequency response of a perturbation type.

The settings are the same as in the parent Prescribed Displacement, Prescribed Displacement, Interface, Prescribed Velocity, Prescribed Velocity, Interface, Prescribed Acceleration, Prescribed Acceleration, Interface, Prescribed Displacement/Rotation or Rigid Connector node. Only degrees of freedom selected as prescribed in the parent node can be assigned a value.

See Harmonic Perturbation in the *Structural Mechanics Modeling* chapter.

SHELL PROPERTIES

This section is only present in the Layered Shell interface. See also The Shell Properties and Interface Selection Sections and the documentation for the corresponding parent nodes.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Prescribed Displacement** > **Harmonic Perturbation**
Solid Mechanics > **Prescribed Velocity** > **Harmonic Perturbation**
Solid Mechanics > **Prescribed Acceleration** > **Harmonic Perturbation**
Solid Mechanics > **Rigid Material** > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Solid Mechanics>Rigid Connector>Harmonic Perturbation
Solid Mechanics, Explicit Dynamics > **Prescribed Displacement** > **Harmonic Perturbation**
Layered Shell > **Prescribed Displacement** > **Harmonic Perturbation**
Layered Shell > **Prescribed Displacement, Interface** > **Harmonic Perturbation**
Layered Shell > **Prescribed Velocity** > **Harmonic Perturbation**
Layered Shell > **Prescribed Velocity, Interface** > **Harmonic Perturbation**
Layered Shell > **Prescribed Acceleration** > **Harmonic Perturbation**
Layered Shell > **Prescribed Acceleration, Interface** > **Harmonic Perturbation**
Membrane > **Prescribed Displacement** > **Harmonic Perturbation**
Membrane > **Prescribed Velocity** > **Harmonic Perturbation**
Membrane > **Prescribed Acceleration** > **Harmonic Perturbation**
Truss > **Prescribed Displacement** > **Harmonic Perturbation**
Truss > **Prescribed Velocity** > **Harmonic Perturbation**
Truss > **Prescribed Acceleration** > **Harmonic Perturbation**
Truss, Explicit Dynamics > **Prescribed Displacement** > **Harmonic Perturbation**
Wire > **Prescribed Displacement** > **Harmonic Perturbation**
Wire > **Prescribed Velocity** > **Harmonic Perturbation**
Wire > **Prescribed Acceleration** > **Harmonic Perturbation**
Multibody Dynamics > **Prescribed Displacement** > **Harmonic Perturbation**
Multibody Dynamics > **Rigid Material** > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Multibody Dynamics > **Rigid Connector** > **Harmonic Perturbation**

Ribbon

Physics tab with **Prescribed Displacement**, **Prescribed Velocity**, **Prescribed Acceleration**, **Prescribed Displacement/Rotation**, or **Rigid Connector** node selected in the model tree:

Attributes > **Harmonic Perturbation**

Stress Linearization

Use the **Stress Linearization** node to define a line (*stress classification line*; *SCL*) along which a linearization of the stress state will be made. A number of result variables are created for each SCL. Add one **Stress Linearization** node for each SCL along which you want to compute linearized stresses.

A **Stress Linearization** node will not affect the solution, unless you manually have coupled any of its variables to other equations. This means that when a **Stress Linearization** node has been added or modified, you do not need to compute a new solution. It is sufficient to perform an **Update Solution** ( ) to make the variables for the new SCL available.

The **Stress Linearization** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

LINEARIZATION

Select what **Type** of linearization to use — **Line** or **Distributed**. For **Line**, the stress is linearized along a single predefined SCL, whereas for **Distributed**, the stress is linearized in a domain. The direction of the SCL is in the distributed case given by the normal of the selected boundaries from which it originates. In principle, an infinite number of SCLs are created in the domain.

When **Line** is selected, the SCL can either be defined by a set of edges or by two points. Control this by setting **Defined by** to either **Edge** or **Two points**.

If the SCL is defined by **Edge**, the **Linearization Line, Edge** selection can consist of several edges, but they are assumed to form a straight line from one free boundary to another.

If the SCL is defined by **Two points**, you can either choose to use geometrical points or arbitrary coordinates within the solid to define the endpoints of the SCL. An SCL is created by connecting the two points with a straight line. To control how to specify

the endpoints, select either **Point** or **Coordinates** in the **Starting point defined by** and **Endpoint defined by** lists.

- For **Points**, select a point in **Linearization Line, Starting Point** or **Linearization Line, Endpoint**.
- For **Coordinates**, enter the coordinate vector of the point.

When **Distributed** is chosen, select entities in **Boundary Selection** from which the distributed SCLs start. The SCLs are then created along the normal to the selected boundaries, and extends through the entities selected in **Domain Selection**.

For 2D and 2D Axisymmetric geometries, no more inputs are need to define the SCL. For 3D, you also need to specify the orientation of the local coordinate system in which the linearized stresses are represented.

SECOND AXIS ORIENTATION

This section is present only in 3D. Here you specify the orientation of the local coordinate system in which the components of the linearized stress tensor are represented. The first direction is always along the SCL, and you indicate the second direction. The third direction is automatically formed by being orthogonal to the SCL and the second direction.

For the **Line** linearization type, select **Reference point** or **Orientation vector**. When a reference point is used, the second local direction will be in the plane formed by the SCL and the selected point. When an orientation vector is used, the second local direction will be directed in the plane formed by the SCL and the selected orientation. The orientation is always adjusted so that it is orthogonal to the SCL.

- For **Reference point**, select **Defined by** — **Point** or **Coordinates**. When **Point** is used, select the reference point in the **Second Axis Orientation Reference Point** section. When **Coordinates** is used, enter the coordinates for the reference point in the **Reference point defining local 2 direction** table.
- For **Orientation vector** select **Defined by** — **Edge** or **Coordinates**. When **Edge** is used, select an edge in the geometry as orientation vector in the **Second Axis Orientation Reference Vector** section. When **Coordinates** is used, enter the orientation vector manually in the **Orientation vector defining local 2 direction** table.

For the **Distributed** linearization type, select a boundary system from the **Defined by** list. The second direction of the SCL is always taken as the first tangent vector of the selected boundary system.

RESULTS

This section is only present for the **Distributed** linearization type. Enter the coordinates for the starting point of a SCL; it must be part of the selected boundaries. The specified point is used to create a **Linearization Line** dataset to plot a single SCL that extends from the point through the selected domains.

Each **Stress Linearization** node will add one plot group under **Result Templates**. It contains line graphs along the stress classification line for the actually computed stress, the membrane stress, and the combined membrane and bending stress.

For details see Results under *Stress Linearization* in the *Structural Mechanics Modeling* chapter.

See also:

- Stress Linearization in the *Structural Mechanics Modeling* chapter.
- Stress Linearization in the *Structural Mechanics Theory* chapter.
- Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Variables** > **Stress Linearization**

Solid Mechanics, Explicit Dynamics > **Variables** > **Stress Linearization**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected in the Model Builder tree:

Global > **Variables** > **Stress Linearization**

Section Forces

Add a **Section Forces** node to compute the integrated section forces on a cross section. The cross section can consist of more than one boundary in a solid, but all the selected

boundaries must be planar and located in the same plane. The section forces are computed in a local coordinate system. The local section forces are denoted as follows:

- `N` — Axial force, normal to the cross section
- `T1` — Shear force along first local axis
- `T2` — Shear force along second local axis
- `Mt` — Twisting moment
- `M1` — Bending moment around first local axis
- `M2` — Bending moment around second local axis

The details are described under Coordinate System Selection below.

The section forces are tabulated in an evaluation group named **Section Forces**.

Two plot groups, **Section Forces** and **Section Moments**, are added to the list of **Result Templates**.

The forces and moments are shown with arrows and their numerical values. The convention for the force arrows is:

- Red: Axial force `N`
- Green: Shear force `T1`
- Blue: Shear force `T2`

The moment arrows have double heads and use the following colors:

- Red: Twisting moment `Mt`
- Green: Bending moment `M1`
- Blue: Bending `M2`

You can disable the **Annotation** node in the corresponding plot group if the annotation box obscures the arrows.

These variables are computed with a higher accuracy when the **Compute boundary fluxes** checkbox is selected in the **Discretization** section. For more information, see Using Surface Traction to Evaluate Reaction Forces and Evaluating Surface Traction Forces on Interior Boundaries in the *Structural Mechanics Modeling* chapter.

You do not need to compute a new solution when a **Section Forces** node is added or modified. It is sufficient to perform an **Update Solution** ( ) to make the variables available. See also Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

CUT PLANE SELECTION

Select the boundaries that define the cross section over which the section forces are to be computed. As a default, it is assumed that the cross section is contiguous. This assumption can be relaxed by selecting the **Allow disconnected boundaries** checkbox under the Cut Plane section.

COORDINATE SYSTEM SELECTION

Select a coordinate system that defines the orientations of the section forces. Only boundary systems with constant axis directions can be selected. The orientation of the local axes are defined as follows:

- The axial force and twisting moment are positive in the direction of the normal of the given boundary system. For an exterior boundary, this is by default the outward normal. For an interior boundary, you can use the **Reverse Normal** subnode under a **Boundary System** to control the orientation.
- If the **Symmetry plane** checkbox is cleared, the first local axis defining the section forces is the same as the first in-plane direction of the selected coordinate system.
- If the **Symmetry plane** checkbox is selected, only the normal orientation of the boundary system is used. The first local axis is taken as the inward normal from the symmetry plane into the structure.
- The second in-plane direction is determined by the cross product $\mathbf{e}_2 = \mathbf{n} \times \mathbf{e}_1$.

CUT PLANE

Select the **Allow disconnected boundaries** checkbox for the case that the cross section is defined by boundaries that are not all adjacent to each other. As a default, this situation will generate a warning message, and no results are computed.

Select the **Symmetry plane** checkbox if the section forces are to be computed for a cross section for which only one half is modeled due to symmetry. When selected, choose a **Symmetry type** — **Symmetry** or **Antisymmetry**, matching the type of symmetry conditions being used. Also, you need to select a boundary in the symmetry plane in the Symmetry Plane Selection section.

Using a symmetry plane has the following effects:

- The orientation of the section forces are controlled by the orientation of the symmetry plane, as described under Coordinate System Selection.
- The center of gravity of the section is assumed to be located in the symmetry plane.
- Section forces will be doubled, so that the values for the complete section are obtained.
- Some of the section forces are set to zero, as dictated by the type of symmetry.

TABLE 4-13: EFFECT OF SYMMETRY SETTINGS ON SECTION FORCES.

TYPE	N	T1	T2	MT	M1	M2
Symmetry	Double	0	Double	0	Double	0
Antisymmetry	0	Double	0	Double	0	Double

SYMMETRY PLANE SELECTION

This section is only visible if the **Symmetry plane** checkbox is selected. In that case, select one boundary in the symmetry plane to identify its location and orientation. The symmetry plane should be orthogonal to the cut plane.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Variables** > **Section Forces**

Solid Mechanics, Explicit Dynamics > **Variables** > **Section Forces**

Ribbon

Physics tab with **Solid Mechanics** or **Solid Mechanics, Explicit Dynamics** selected in the Model Builder tree:

Global > **Variables** > **Section Forces**

Test Material

Under large strains, several material models have regions where the material is unstable. These instabilities might be triggered by a uniaxial deformation limit, or by a combination of multiaxial deformations. These instabilities indicate that the material is used outside the validity range where the material parameters were obtained. Unstable materials generate convergence problems and unphysical solutions, and therefore it is beneficial to test a given material model before running big models.

Some options described below are only available with certain COMSOL products (see www.comsol.com/products/specifications/).

DOMAIN SELECTION

You can only select one domain, so that one material model can be tested at the time. If you want to test different material models, you can sequentially change the domain selection, or add several **Test Material** nodes with different selections.

MATERIAL TESTS

Select the **Specimen size** — **Automatic** or **User defined**. For **Automatic**, a single mesh element with a length of 1 m is used. Select **User defined** to specify a different **Characteristic length** L in case the specimen size has an influence in the material law.

Select the **Study setup** — **Stationary** or **Time dependent**, and then select the **Test setup** — **Monotonic**, **Cyclic**, or **User defined**.

For the **Time dependent** test, enter the **Test time** t_f. The default is 1 second.

For either the **Stationary** or the **Time dependent** test, enter the **Number of measurement points in the study** N_p. The default is 50 points.

For the **Monotonic** test, the number of measurement points sets up the resolution for the given load or deformation interval.

For the **Cyclic** test, enter the **Number of cycles** N_c. In a **Stationary** test, each cycle has N_p/N_c measurement points per cycle. In a **Time dependent** test, each cycle has a period of $t_c = t_f/N_c$, and a time step given by $\Delta t = t_f/N_p$.

For the **User defined** test, select the **Test control** — **Stretch driven** or **Force driven**. Depending on the choice, you can specify how the prescribed displacement or load varies as a function of time or parameter range. Enter a **Parameter name** (default is `para`) to be used in the user defined inputs for the stretch functions or force functions. A parameter with the given name will be created in the Parameters node when clicking **Set Up and Run Tests** ( ) as shown below.

Tests

Select one or more of the available checkboxes to test the material under different load conditions:

- **Uniaxial test** — Enter the stretch range for the test. The minimum stretch λ_{min} gives the compressive limit, and the maximum stretch λ_{max} gives the tensile limit. A

uniaxial compression test is achieved by setting $\lambda_{max} = 1$, and a uniaxial extension test by setting $\lambda_{min} = 1$. The inputs should be such $\lambda_{min} < \lambda_{max}$.

- **Biaxial test** — Enter the minimum stretch λ_{min}, the maximum stretch λ_{max}, and the strain biaxiality ratio β for the test.
- **Shear test** — Enter the maximum shear angle θ for the test.
- **Isotropic test** — Enter the minimum stretch λ_{min} and the maximum stretch λ_{max} for the test.
- **Oedometer test** — Enter the axial stretch λ_{min} for the test.
- **Triaxial test** — Enter the in situ stress σ_{ins} and the axial stretch λ_{min} for the test.

Isotropic Compression with Modified Cam-Clay Material Model: Application Library path **Geomechanics_Module/Verification_Examples/isotropic_compression**.

Triaxial and Oedometer Test with Modified Cam-Clay Material Model: Application Library path **Geomechanics_Module/Verification_Examples/isotropic_compression_mscc**.

Load Group, Material, and Study Generation

The material parameters of a given material model are tested by creating dedicated tests on a single element. In order to do this, a number of fundamental load cases are analyzed. These steps are described below.

On the **Material Tests** section toolbar, there is an icon **Automated Model Setup** ( ). Select:

- **Set Up and Run Tests** ( ). This option generates a new 3D Component, which contains a single element to test the material model defined under **Domain Selection**. When you select it, the following changes will be made to the model:
  - A new 3D Component is created.
  - A new **Solid Mechanics** interface is added to this component.
  - A number of boundary conditions will be available to the new **Solid Mechanics** interface. The boundary conditions correspond to the selected tests.
  - A **Mesh** node containing a single element.
  - A new **Study** node will be added based on the **Study setup** selection. A stationary or time dependent study will contain the **Number of measurement points** for either

a **Monotonic** or a **Cyclic** test. This study runs automatically after selecting **Set Up and Run Tests**.

- A **Stop Condition** node will be added under the stationary or time dependent study. This condition stops the loading when the element is fully collapsed. See Stop Condition in the *COMSOL Multiphysics Reference Manual*.
- A new **Material Tests** node will be added to the **Results** node. Different plots will show the **Tests** and the results from the **Study** node.

- **Remove Tests** ( ). This option deletes the 3D Component, the Solid Mechanics interface, and the Mesh, Study, and Results nodes added after selecting **Set Up and Run Tests**.

Local System Results

Add a **Local System Results** node to create variables for common structural mechanics quantities having vector and tensor components oriented in a local coordinate system.

When a **Local System Results** node has been added, you do not need to compute a new solution. It is sufficient to perform an **Update Solution** ( ) to make the new variables available for postprocessing.

COORDINATE SYSTEM SELECTION

Select a coordinate system defined under **Definitions** from the list. Only coordinate systems with orthonormal axis directions will be shown.

TABLE 4-14: QUANTITIES TRANSFORMED BY LOCAL SYSTEM RESULTS.

QUANTITY	VARIABLE NAME	EXAMPLE
Displacement	<interface>.<tag>.<dofname>i	`solid.lsr1.u1`
Velocity	<interface>.<tag>.<dofname>_ti	`solid.lsr1.u_t1`
Acceleration	<interface>.<tag>.<dofname>_tti	`solid.lsr1.u_tt1`
Cauchy Stress	<interface>.<tag>.slij	`solid.lsr1.sl12`
Total strain (Green–Lagrange)	<interface>.<tag>.elij	`solid.lsr1.el12`
Elastic strain (Green–Lagrange)	<interface>.<tag>.eelij	`solid.lsr1.el12`
Elasticity matrix	<interface>.<tag>.Dij	`solid.lsr1.Dl12`

- If you have selected a local coordinate system in a material model, stress and strain tensors in the corresponding local directions are always available.
- For more general transformations, and to transform other quantities than those supplied by **Local System Results**, you can use Vector Transform or Matrix Transform.

Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Variables** > **Local System Results**
Solid Mechanics, Explicit Dynamics > **Variables** > **Local System Results**
Multibody Dynamics > **Variables** > **Local System Results**
Solid Rotor > **Variables** > **Local System Results**

Ribbon

Physics tab with **Solid Mechanics**; **Solid Mechanics, Explicit Dynamics**; **Multibody Dynamics**; or **Solid Rotor** selected in the Model Builder tree:

Global > **Variables** > **Local System Results**

Average Rotation

Add an **Average Rotation** node to compute the average rotation and displacement for a set of points. The points are assumed to move together as a rigid body in a least-squares sense. Vectors for average rotation and displacement, as well as their velocities and accelerations, are created as result variables. Use one **Average Rotation** node for each set of points that you want to evaluate.

The point selection requires at least two points in 2D and three points in 3D.

CENTER OF ROTATION

Select a method for determining the **Center of Rotation** — **Automatic**, **Centroid of selected entities**, or **User defined**.

- If **Automatic** is selected, the center of rotation is determined as the average position of the selected points.
- For **Centroid of selected entities**, select an **Entity level** — **Domain**, **Boundary**, **Edge**, or **Point**. The available choices depend on physics interface and geometrical dimension. The center of rotation is located at the centroid of the selected entities, which do not need to be related to the points used for computing the average rotation. As a special case, you can select a single point, and thus use that point as center of rotation.

Once chosen, a default Center of Rotation: Domain, Center of Rotation: Boundary, Center of Rotation: Edge, or Center of Rotation: Point subnode is automatically added.

- When **User defined** is selected, enter the location of the center of rotation manually.

ROTATION MODEL

Select a rotation model — **Small rotations**, **Finite rotations, symmetric segregated**, or **Finite rotations, unsymmetric coupled**.

With **Small rotations**, the computation of displacements caused by rotations is linearized using a cross product. This means that the average rotation can be evaluated using explicit expressions, so the only effect of using a **Average Rotation** node is to define some new variables.

The finite rotation options, however, require the solution of an extra set of nonlinear equations. There is still no influence on the structural mechanics solution as such. Using an appropriate choice of finite rotation formulation, and a corresponding setup of the solver strategy can strongly affect the computational cost. The cost of solving for the average finite rotation variables as such is very small, but the addition of these degrees of freedom can affect the structure of the system of equations.

With the unsymmetric coupled option, you do not have to care about solver setup. However, a majority of structural mechanics problems produce a symmetric stiffness matrix, and that symmetry is now broken by adding the equations for the average rotation. This will lead to a significant (about a factor 2) penalty on solution time and memory usage.

If the variables created by this feature are used in other equations, you should also typically use the unsymmetric coupled option.

In most cases, the average rotation is however used only during result presentation. This means that the most efficient solution procedure is to first compute the structural mechanics displacements, and subsequently solve the least squares problem for the average rigid rotation. If this is the case, using the symmetric segregated approach is usually the best choice. You must, however, set up the solver sequence in a way such that you can benefit from this property. There are several possible strategies, for example:

- Use two different study steps. In the first step, do not solve for the average rotation variables. This can be done by disabling the **Average Rotation** node in the study settings. In the second study step, solve only for those variables by disabling the other dependent variables under the **Dependent Variables** node.
- A similar approach is to use a separate study, rather than a study step, for computing the average rotation. The main difference is that you manually have to connect the two studies using the **Values of Dependent Variables** section of the second study.
- Use a segregated solver. In this case, solve for the average rotation variables only in the last segregated step. Note that in the default case, always at least two iterations will be performed in a segregated solver. This is typically not what you want. Set **Termination technique** to **Iterations** in the **Segregated** node, and terminate after one iteration. Then, if required, set up the **Segregated Step** in which the structural mechanics problem is solved so that proper nonlinear iterations are performed until convergence.

See also Average Rotation in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Variables** > **Average Rotation**
Membrane > **Variables** > **Average Rotation**
Truss > **Variables** > **Average Rotation**
Wire > **Variables** > **Average Rotation**
Multibody Dynamics > **Global** > **Average Rotation**
Solid Rotor > **Variables** > **Average Rotation**

Ribbon

Physics tab with **Solid Mechanics**, **Membrane**, **Truss**, **Wire**, or **Solid Rotor** selected:

Global > **Variables** > **Average Rotation**

Physics tab with **Multibody Dynamics** selected:

Global > **Multibody Dynamics** > **Average Rotation**

Warpage

Add a **Warpage** node to evaluate the change in shape of a surface. Warpage is defined as the difference between the actual deformation of a surface and an average rigid motion of the same surface or a reference plane. Warpage is measured along the normal of the surface.

There is no limitation in the selection of the boundaries forming the surface, but in most cases, this feature is used to measure how much an originally flat surface deviates from being planar after deformation.

In most cases, a **Warpage** node will not affect the solution. It will merely add variables for result evaluation. When a **Warpage** node has been added or modified, you then do not need to compute a new solution. It is sufficient to perform an **Update Solution** ( ) to make the variables for result evaluation available.

Exceptions to this, making a new solution necessary, is when:

- One of the finite rotation models for warping is selected under **Rotation Model**.
- You manually couple warpage variables to some other equation.

The **Warpage** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

ROTATION MODEL

Select a **Rotation model** — **Small rotations**, **Finite rotations, symmetric segregated**, or **Finite rotations, unsymmetric coupled**.

With the small rotation formulation, warping can be evaluated using explicit expressions, so the only effect of using a **Warpage** node is to define new variables.

The finite rotation options, however, require solving an extra set of nonlinear equations. There is still no influence on the structural mechanics solution as such. Using an appropriate choice of finite rotation formulation and a corresponding setup of the solver strategy can strongly affect the computational cost. The cost of solving for the average finite rotation variables is very small, but the addition of these extra degrees of freedom can affect the structure of the system of equations.

With the unsymmetric coupled option, you do not have to care about solver setup. However, a majority of structural mechanics problems produce symmetric stiffness matrices, and that symmetry is broken when adding the equations for the average rotation. This will lead to a significant (about a factor 2) penalty on solution time and memory usage.

If the variables created by **Warpage** are used in other equations, you should also typically use the unsymmetric coupled option.

In most cases, warping is however used only during result visualization. This means that the most efficient procedure is to first compute the structural mechanics displacements, and subsequently solve the least squares problem for computing the surface's average rigid rotation. For this case, using the symmetric segregated approach is usually the best choice. You must, however, set up the solver sequence in a way such that you can benefit from this property. There are several possible strategies, for example:

- Use two different study steps. In the first step, do not solve for the warpage average rotation variables. This can be done by disabling the **Warpage** node in the study settings. In the second study step, solve only for those variables by disabling the other dependent variables under the **Dependent Variables** node.
- A similar approach is to use a separate study, rather than a study step, for computing the average rotation. The main difference is that you manually have to connect the two studies using the **Values of Dependent Variables** section of the second study.
- Use a segregated solver. In this case, solve for the warpage average rotation variables only in the last segregated step. Note that in the default case, always at least two iterations will be performed in a segregated solver. This is typically not what you want. Set **Termination technique** to **Iterations** in the **Segregated** node, and terminate after one iteration. Then, if required, set up the **Segregated Step** in which the structural mechanics problem is solved so that proper nonlinear iterations are performed until convergence.

REFERENCE PLANE

Select a **Reference plane** — **Automatic** or **From points**.

With the **Automatic** formulation, warping is evaluated using an average rigid motion of the selected surface.

When **From points** is selected, specify the points that define the reference plane. In case the **Symmetry type** is set to **Symmetry** or **Antisymmetry**, the points should not be placed in the symmetry plane.

SYMMETRY TYPE

Select a **Symmetry type** — **None**, **Symmetry**, or **Antisymmetry**.

With the **Symmetry** and **Antisymmetry** formulations, warping evaluation is compensated to include the symmetry or antisymmetry condition, respectively. These formulations require that you specify the entity that represents the symmetry plane in the **Symmetry Plane Selection** section. The **Antisymmetry** formulation is only available when **Rotation model** is set to **Small rotations**.

See also Warpage in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Variables** > **Warpage**
Shell > **Variables** > **Warpage**
Layered Shell > **Variables** > **Warpage**

Ribbon

Physics tab with an applicable physics interface selected in the Model Builder tree:

Global > **Variables** > **Warpage**

Wave Speeds

When you add a **Wave Speeds** node, a number of postprocessing variables containing information about wave speeds in the material are created. This node does not influence the solution.

When a **Wave Speeds** node has been added, you do not need to compute a new solution. It is sufficient to perform an **Update Solution** ( ) to make the new variables available for result presentation.

The **Wave Speeds** node is only available with some COMSOL products (see www.comsol.com/products/specifications/). It is available for 3D, 2D, and 2D axisymmetry.

WAVE SPEEDS

Select a **Wave propagation direction** — **Coordinate system axes** or **User Defined**. This is the propagation direction with respect to which the wave speeds will be computed.

When is **Coordinate system axes** used, the wave propagation direction is along the first axis of the selected coordinate system.

When **User Defined** is selected, enter also the **Direction vector**, $\mathbf{n}$. The vector, which is interpreted in the selected coordinate system, gives the wave propagation direction.

Select **Compute group velocity** in order to create variables also for the group velocities. The default is to only compute phase velocities.

See also

- Wave Speed Computation for the theory for a linear elastic material.
- Wave Propagation in Piezoelectric Media for the theory for a piezoelectric material.
- Updating a Solution in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Linear Elastic Material** > **Variables** > **Wave Speeds**
Solid Mechanics > **Piezoelectric Material** > **Variables** > **Wave Speeds**
Solid Mechanics, Explicit Dynamics > **Linear Elastic Material** > **Variables** > **Wave Speeds**

Ribbon

Physics tab with **Linear Elastic Material** or **Piezoelectric Material** selected in the Model Builder tree:

Attributes > **Wave Speeds**

Important Variables in the Solid Mechanics Interface

In each node in the physics interface, a number of variables are defined. In Table 4-15, you can find a list of variables that you as a user may want to use when, for example, creating your own variables and expressions. This list is far from exhaustive but contains some of the most commonly used variables. To see all variables defined by a certain node in the Model Builder tree, enable Equation View, and examine the contents in the **Equation View** node under each physics node in the Model Builder tree.

The scopes of the variables in the table serve as examples only and assume the default first instance of a certain feature in the Model Builder tree. The actual scope will depend on the tag of a certain node in the model tree.

In the case of vectors and tensors, only a single component is shown. The actual indices can depend on the dimensionality of the problem and names of coordinate system axis directions.

Variables created in contact analysis are summarized in Dependent Variables in Contact Analysis and Important Contact Variables

In the *COMSOL Multiphysics Reference Manual*:

- For more information about the **Equation View** node, see Equation View.
- For more information about variable scopes, see Variable Naming Convention and Namespace.
- For more information about tags in the model tree, see Displaying Node Names, Tags, and Types in the Model Builder.

TABLE 4-15: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
`solid.disp`	Total displacement	All materials[1]	
`solid.vel`	Velocity magnitude	All materials[1]	
`solid.u_tX`	Velocity, X component	All materials[1]	
`solid.acc`	Acceleration magnitude	All materials[1]	
`solid.u_ttX`	Acceleration, X component	All materials[1]	

TABLE 4-15: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
`solid.eXX`	Strain tensor, XX component	All materials[1]	Total (Green–Lagrange) strain in global directions. See also Deformation Measures.
`solid.el11`	Strain tensor, local coordinate system, 11 component	All materials[1]	Total strain in the local directions.
`solid.eel11`	Elastic strain tensor, local coordinate system, 11 component	All materials[1]	
`solid.eeldev11`	Deviatoric elastic strain tensor, local coordinate system, 11 component	All materials[1]	
`solid.eiel11`	Inelastic strain tensor, local coordinate system, 11 component	All materials[1]	See also Inelastic Strain Contributions.
`solid.FdxX`	Deformation gradient, xX component	All materials[1]	See also Deformation Measures.
`solid.Fdlx1`	Deformation gradient, local, x1 component	All materials[1]	
`solid.J`	Volume ratio	All materials[1]	Total relative volume change.
`solid.Cel11`	Elastic Cauchy–Green tensor, local coordinate system, 11 component	All materials[1]	
`solid.Jel`	Elastic volume ratio	All materials[1]	Elastic relative volume change.
`solid.Ldxy`	Rate of strain tensor, xy component	All materials[1]	See also Strain Rate and Spin.
`solid.Lwxy`	Spin tensor, xy component	All materials[1]	See also Strain Rate and Spin.

TABLE 4-15: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
`solid.SXX`	Second Piola–Kirchhoff stress, XX component	All materials[1]	See also Defining Stress.
`solid.Sl11`	Second Piola–Kirchhoff stress, local coordinate system, 11 component	All materials[1]	
`solid.SdevX`	Deviatoric second Piola–Kirchhoff stress, X component	All material[1]	See also Invariants of the Stress Tensor.
`solid.PxX`	First Piola–Kirchhoff stress, xX component	All materials[1]	See also Defining Stress.
`solid.sxy`	Stress tensor, xy component	All materials[1]	Cauchy stress, See also Defining Stress.
`solid.sdevx`	Deviatoric stress tensor, x component	All materials[1]	Deviatoric Cauchy stress, See also Invariants of the Stress Tensor.
`solid.mises`	von Mises stress	All materials[1]	See also Invariants of the Stress Tensor.
`solid.tresca`	Tresca stress	All materials[1]	See also Invariants of the Stress Tensor.
`solid.I1s`	First principal invariant of stress	All materials[1]	See also Invariants of the Stress Tensor.
`solid.I2s`	Second principal invariant of stress	All materials[1]	See also Invariants of the Stress Tensor.
`solid.I3s`	Third principal invariant of stress	All materials[1]	See also Invariants of the Stress Tensor.
`solid.II2s`	Second invariant of stress deviator	All materials[1]	See also Invariants of the Stress Tensor.

TABLE 4-15: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE.

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
`solid.II3s`	Third invariant of stress deviator	All materials[1]	See also Invariants of the Stress Tensor.
`solid.RFx`	Reaction force, x component	All materials[1]	
`solid.RMx`	Reaction moment, y component	All materials[1]	
`solid.Tracx`	Nominal traction (force/area), x component	All materials[1]	
`solid.Tax`	Traction (force/area), x component	All materials[1]	

1 – All materials: **Linear Elastic Material**, **Nonlinear Elastic Material**, **Hyperelastic Material**, **Piezoelectric Material**, **Piezomagnetic Material**, **Elastoplastic Soil Material**, **Shape Memory Alloy**.

5

Solid Mechanics, Explicit Dynamics

This chapter describes the Solid Mechanics, Explicit Dynamics ( ) interface, which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Solid Mechanics, Explicit Dynamics Interface

The **Solid Mechanics, Explicit Dynamics (solid)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is intended for general time explicit analysis of 3D, 2D plane strain, or 2D axisymmetric bodies. The Solid Mechanics, Explicit Dynamics interface is based on solving the equations of motion together with a constitutive model for a solid material. Results such as displacements, stresses, and strains are computed.

	Explicit Dynamics in the COMSOL Multiphysics Reference Manual.

This interface is similar to the The Solid Mechanics Interface, upon which it is based. There are a number of specializations for explicit dynamics, but the interface can also be used with most other study types. The most important use case for that is to provide initial conditions for a time-explicit analysis.

Some important differences, when compared to the Solid Mechanics interface are:

- The only two top level material models available are **Linear Elastic Material** and **Hyperelastic Material**. Most subnodes under these materials, such as **Plasticity**, **Damping**, or **Fiber** are available.
- Some other features are not available. Most noticeably: **Rigid Connector**, **Thin Layer**, and **Spring-Damper**. A detailed list of the available features can be found in the section Domain, Boundary, Edge, Point, Global, and Pair Nodes for Solid Mechanics, Explicit Dynamics below.
- Only linear shape functions are supported. Reduced integration is used as a default.
- Weak constraints are not available.
- In cases where there are smaller differences between the settings for a certain feature, for example different options is a list, it is noted in the documentation of that feature.

Note that you can move nodes between the Solid Mechanics and the Solid Mechanics, Explicit Dynamics interfaces simply by drag-and-drop.

The default material is a Linear Elastic Material. With either the Nonlinear Structural Materials Module or the Geomechanics Module, the physics interface is extended with more materials, for example, material models for plasticity, creep, and concrete.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, the following default nodes are also added to the **Model Builder**: **Linear Elastic Material**, **Free** (a boundary condition where boundaries are free, with no loads or constraints), and **Initial Values**. For axisymmetric models, an **Axial Symmetry** node is also added.

Then, from the **Physics** toolbar, you can add other nodes that implement, for example, solid mechanics material models, boundary conditions, and loads. You can also right-click **Solid Mechanics, Explicit Dynamics** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `solid`.

2D APPROXIMATION

The **2D approximation** list contains only one entry, **Plane strain**,

THICKNESS

For 2D components, enter a value or an expression for the thickness d. The default value of 1 m is suitable for plane strain models, where it represents a unit-thickness slice, for example.

When manually combining Solid Mechanics, Explicit Dynamics with other physics interfaces, you must make sure that the same thickness assumption is used everywhere. In most cases, the default settings will be correct since interfaces that do not have an explicit thickness property will implicitly assume unit thickness. When using other unit systems than the default SI system, you need to pay special attention to the values of this property.

Use a Change Thickness node to change thickness in parts of the geometry if necessary.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasistatic (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system.

ENERGY DISSIPATION

From the **Store dissipation** list, select **Individual contributions**, **Total**, or **Off**.

Select **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Select **Total** to accumulate all the dissipative processes into one variable.

DISCRETIZATION

In the Solid Mechanics, Explicit Dynamics interface, the only available discretization is linear shape functions.

To display other settings for this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

DEPENDENT VARIABLES

The physics interface uses the global spatial components of the **Displacement field** u as dependent variables. The default names for the components are (u, v, w) in 3D. In 2D the component names are (u, v), and in 2D axisymmetry they are (u, w). You can however not use the 'missing' component names in 2D as a parameter or variable name, since they are used internally.

You can change both the field name and the individual component names. If a new field name coincides with the name of another displacement field, the two fields (and the interfaces that define them) share degrees of freedom and dependent variable component names. A new field name must not coincide with the name of a field of another type (that is, it must contain a displacement field), or with a component name

belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

When a Solid Mechanics, Explicit Dynamics interface is active in the model, a set of default and predefined result features are created.

- A default plot named **Stress (solid)** containing the von Mises equivalent stress is created.
- In the case of 2D axisymmetry, additional default plots are created using a revolved 3D representation. These plots are named **Stress (solid), 3D** and **Mode Shape (solid), 3D** respectively.
- A result template named **Displacement (solid)** containing the norm of the displacement vector is created.
- A result template named **Velocity (solid)** containing the norm of the velocity vector is created when either a time-domain or frequency-domain type of study is used.
- A result template named **Maximum Principal Tensile Stress (solid)** containing the largest principal stress is created.
- A result template named **Maximum Principal Compressive Stress (solid)** containing the smallest principal stress is created.
- A number of result templates containing vector plots of various types of loads are created. The number of load plots depends on the types of load features that are active in the Solid Mechanics, Explicit Dynamics interface. The naming of the load groups follows the pattern **Boundary Loads (solid)**, **Point Loads (solid)**, and so on. They are all placed in a group node named **Applied Loads (solid)**.

There are also several nodes in the Solid Mechanics, Explicit Dynamics interface that can trigger the generation of specialized default or predefined result features. This is documented under the respective nodes.

See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

- Domain, Boundary, Edge, Point, Global, and Pair Nodes for Solid Mechanics, Explicit Dynamics
- Solid Mechanics Theory

- *Bracket — Explicit Dynamics*
- *Objects Falling in a Box*

Domain, Boundary, Edge, Point, Global, and Pair Nodes for Solid Mechanics, Explicit Dynamics

Most of the features in the Solid Mechanics, Explicit Dynamics interface are identical to the same features in the Solid Mechanics interface, and are thus documented there.

The domain, boundary, edge, point, and pair nodes and subnodes (listed in alphabetical order), which are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Solid Mechanics, Explicit Dynamics interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Domain** level are **Material Models**; **Volume Forces**; **Mass, Spring, and Damper**; **Domain Constraints**; **Variables**; and **More**.

The submenus at the **Boundary** level are **Mass, Spring, and Damper**; **More Constraints**; **Pairs**; and **More**.

There are also the **Edges** and **Points** submenus.

LINKS TO FEATURE NODE INFORMATION

These nodes are described in this section (listed in alphabetical order)

- Artificial Viscosity
- Mass Scaling
- Prescribed Displacement (on Domains)

See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

These following top level nodes are described in the documentation for the Solid Mechanics interface.

Note: Some features are only present with certain COMSOL products.

- Added Mass
- Adiabatic Heating
- Antisymmetry
- Bolt Pretension
- Bolt Thread Contact
- Body Load
- Boundary Load
- Contact
- Continuity
- Crack
- Edge Load
- Fiber
- Fixed Constraint
- Free
- General Contact
- Gravity
- Hyperelastic Material
- Initial Values
- Interior Contact
- Linear Elastic Material
- Linearly Accelerated Frame
- Local System Results
- Low-Reflecting Boundary
- Point Load
- Point Load (on Axis)
- Point Mass
- Prescribed Displacement (on points, edges, and boundaries)
- Ring Load
- Roller
- Rotating Frame
- Section Forces
- Symmetry
- Symmetry Plane
- Spring Foundation
- Stress Linearization
- Wave Speeds

Below is an alphabetic list of subnodes, documented for the Solid Mechanics interface.

- Activation
- Adhesion
- Bolt Selection
- Concrete
- Contact
- Contact Model
- Crack Closure
- Creep
- Damping
- Damping (General Contact)
- Damage
- Decohesion
- External Strain
- External Stress
- Face Load (Crack)
- Fiber
- Friction
- Friction (General Contact)
- Harmonic Perturbation
- Hygroscopic Swelling
- Inelastic Strain Rate
- Initial Stress and Strain
- J-Integral
- No Contact
- Offset
- Phase
- Plasticity
- Predeformation
- Pressure-Dependent Plasticity
- Reverse Crack Front
- Safety
- Soil Plasticity
- Stabilization
- Thermal Expansion (Fiber)
- Thermal Expansion (for Materials)
- Virtual Crack Extension
- Viscoelasticity
- Viscoplasticity
- Wear

Time Dependent in the *COMSOL Multiphysics Reference Manual*.

See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Prescribed Displacement (on Domains)

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in the selected domain.

COORDINATE SYSTEM SELECTION

The constraints are applied in the axis directions of the coordinate system selected in this section. You should only use a coordinate system that has fixed spatial directions.

PRESCRIBED DISPLACEMENT

Constrain the displacement component to a given value. Enter a scalar value for the components of the prescribed displacement u_0.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings
- Excluded Surfaces, Excluded Edges, and Excluded Points

When **Individual dependent variables** is selected in the **Apply reaction terms on** list, the constraint forces are applied directly on the degrees of freedom, which are the displacements along the global coordinate axes. If you use this setting together with a local coordinate system, the results will be inconsistent since the constraint forces will not match the constraint orientation.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics, Explicit Dynamics > **Domain Constraints** > **Prescribed Displacement**

Truss, Explicit Dynamics > **Line Constraints** > **Prescribed Displacement**

Ribbon
Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Domains > **Domain Constraints** > **Prescribed Displacement**

Physics tab with **Truss, Explicit Dynamics** selected:

Edges > **Line Constraints** > **Prescribed Displacement**

Mass Scaling

Use the **Mass Scaling** subnode to increase the density in small mesh elements in the domain.

The mass scaling method is used to increase the critical (stable) time increment in explicit dynamics studies.

When the density and stiffness are homogeneous, the smallest mesh elements limit the time step as the critical time step is proportional to the mesh element size h. Hence, a single mesh element with poor quality can severely degrade the step size and ultimately the performance of the explicit dynamic analysis.

Enter the **Target cell time** to automatically increase the density in such small elements.

LOCATION IN USER INTERFACE

Context Menus
Solid Mechanics, Explicit Dynamics > **More** > **Mass Scaling** (Domain)
Truss, Explicit Dynamics > **More** > **Mass Scaling** (Edge)

Ribbon
Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Domains > **More** > **Mass Scaling**

Physics tab with **Truss, Explicit Dynamics** selected:

Edges > **More** > **Mass Scaling**

Artificial Viscosity

Use the **Artificial Viscosity** subnode to add damping to the material model. Artificial viscosity can be used in the Explicit Dynamics study; for other study types the settings in the **Artificial Viscosity** subnode are ignored.

ARTIFICIAL VISCOSITY

Select the **Method** and enter the settings accordingly. The material will get additional stresses proportional to the strain rate.

Deviatoric

The additional stress is proportional to the deviatoric strain rate. Enter the **Linear coefficient** c_L and the **Quadratic coefficient** c_Q to define the shear viscosity η_v.

Volumetric

The additional stress is proportional to the volumetric strain rate. Enter the **Linear coefficient** c_L and the **Quadratic coefficient** c_Q to define the bulk viscosity η_b.

See also Viscous Damping in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics, Explicit Dynamics > **More** > **Artificial Viscosity** (Domain)

Ribbon

Physics tab with **Solid Mechanics, Explicit Dynamics** selected:

Domains > **More** > **Artificial Viscosity**

6

Shell and Plate

This chapter describes the Shell ( ), and Plate ( ) interfaces, which are found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Shell and Plate Interfaces

The **Shell (shell)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used to model structural shells on boundaries in 3D or 2D axisymmetry. Shells are thin flat or curved structures, having significant bending stiffness. The interface uses shell elements of the MITC type, which can be used for analyzing both thin (Kirchhoff theory) and thick (Mindlin theory) shells.

The **Plate (plate)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, provides the ability to model structural plates in 2D. Plates are thin flat structures with significant bending stiffness, being loaded in a direction out of the plane.

The **Shell** interface is for 3D and 2D axisymmetry models.

The **Plate** interface is for 2D models — domains are selected instead of boundaries, and boundaries instead of edges. Otherwise the **Settings** windows are similar to those for the **Shell** interface.

The Linear Elastic Material is the default material model. It adds a linear elastic equation for the displacements and has a Settings window to define the elastic material properties. With this material model, the material is assumed to be homogeneous through the thickness of the shell.

When this interface is added, these default nodes are also added to the **Model Builder** — **Linear Elastic Material**, **Thickness and Offset**, **Free** (a boundary condition where edges are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **Shell** or **Plate** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to

different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `shell` or `plate`.

SKETCH

In the **Sketch** section, a conceptual sketch of the degrees of freedom in the Shell and Plate interfaces appears.

AXIAL SYMMETRY APPROXIMATION

	Select **Circumferential mode extension** to prescribe a circumferential wave number to be used in eigenfrequency or frequency-domain studies. When selected, enter the **Azimuthal mode number** m. For more information, see Circumferential Displacement and Out-of-Plane Waves in the *Structural Mechanics Theory* chapter.

	Eigenfrequency Analysis of a Free Cylinder: Application Library path **Structural_Mechanics_Module/Verification_Examples/free_cylinder**.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

For problems with creep, and sometimes viscoelasticity, the problem can be considered as quasistatic. This is also the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

ENERGY DISSIPATION

From the **Store dissipation** list, select **Individual contributions**, **Total**, or **Off**.

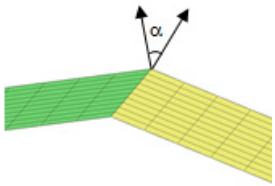
Select **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Select **Total** to accumulate all the dissipative processes into one variable.

FOLD-LINE SETTINGS

This section is available for the Shell interface only. Also see The MITC Shell Formulation.

The fold-line limit angle α is the smallest angle between the normals of two boundaries that makes their intersection to be treated as a fold line. The normal to the shell is discontinuous along a fold-line. Enter a value or expression in the α field. The default value is 0.001 radians (approximately 0.06 degrees). The value must be larger than 0, and less than $\pi/2$. Angles larger than a few degrees are usually not meaningful.



Since the rotational degrees of freedom have different meanings across a fold line, they are treated as separate degrees of freedom connected by a constraint. This constraint is implemented as a pointwise constraint by default.

Select the **Fold-line constraint** — **Pointwise constraint** or **Weak constraint**. Switching to weak constraints may be necessary when weak constraints are used for other boundary conditions.

When a pointwise constraint is used, you can also select the **Constraint method** — **Elemental** or **Nodal**. In the nodal method, one constraint per node is generated. When the elemental method is used, one constraint is generated per node in each element and the duplicates are removed during the constraint elimination pass. In most cases, the nodal method is preferable.

DEFAULT THROUGH-THICKNESS RESULT LOCATION

Enter a number between -1 and 1 for the **Local z-coordinate [-1,1] for thickness-dependent results** Z. The value can be changed from −1 (downside) to +1 (upside). The default is +1. A value of 0 means the midsurface of the shell. This is the default position for stress and strain evaluation during the results analysis. During the

results and analysis stage, the coordinates can be changed in the **Parameters** section in the result features.

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. Normally these settings do not need to be changed.

The **Use MITC interpolation** checkbox is selected by default, and this interpolation, which is part of the MITC shell formulation, should normally always be active.

For the Plate interface, the **Use 3D formulation** checkbox is used to select whether six or three variables are used in the formulation. For geometrically nonlinear analyses, or when in-plane (membrane) forces are active, six variables must be used. This checkbox is selected by default.

As a default, a selective integration scheme is used for the weak contributions used to form the stiffness matrix in 2D axisymmetry. The membrane part of the strain energy is integrated using reduced integration in order to remove locking effects for thin shells. Clear the **Use reduced integration scheme** checkbox to use full integration instead.

In order to maintain the property that the shell normal has unit length, a constraint is applied on the shell normal displacement degrees of freedom in each node. This constraint is, as default, implemented as a pointwise constraint.

Select a **Normal constraint type** — **Pointwise constraint** or **Weak constraint**. Switching to weak constraints may be necessary if weak constraints are used for other boundary conditions. A mesh vertex cannot have a combination of pointwise and weak constraints.

When a pointwise constraint is used, you can also select a **Constraint method** — **Nodal** or **Elemental**. In the nodal method, one constraint per node is generated. When the elemental method is used, one constraint is generated per node in each element and the duplicates are removed during the constraint elimination pass. In most cases, the nodal method is preferable. Not only are fewer constraints generated, but also the risk that the constraints are conflicting due to variations of the normal direction in adjacent elements is eliminated.

You can choose how extra ODE variables added by some features are grouped in the **Dependent Variables** node of a generated solver sequence.

- Select the **Rigid materials** checkbox to group variables added by Rigid Material nodes.
- Select the **Rigid connectors** checkbox to group variables added by Rigid Connector nodes.
- Select the **Attachments** checkbox to group variables added by Attachment nodes.

The selections made here can be overridden by the settings in the **Advanced** section of the Rigid Material, Rigid Connector or Attachment features.

AUTOMATED MODEL SETUP

This section will only be displayed if a mesh on NASTRAN® format, containing RBE2 elements, has been imported in an **Import** node under **Mesh**. The purpose is to automatically create rigid connectors from RBE2 elements in the NASTRAN file.

An RBE2 element represents a rigid connection between a set of mesh vertices. This means that it can, and often does, connect elements from different physics interfaces.

In the menu in the section title, you can select **Create Rigid Connectors from RBE2**. The effect is that one rigid connector will be created for each RBE2 element in the imported file. This will happen for all physics interfaces in the **Interfaces** list. Supported interfaces are: Solid Mechanics, Shell, Beam, and Multibody Dynamics. If there are RBE2 elements spanning more than one physics interface, they will be automatically connected.

The created rigid connectors will have point, edge, and boundary selections as inferred from the nodes in the RBE2 element and the mesh connectivity. The 'independent node' of the RBE2 element is used as center of rotation for the rigid connector.

The **Automated Model Setup** section is present in the Solid Mechanics, Shell, and Beam interfaces. In a model that contains several physics interfaces, you should use the automated model setup from only one of them, and make sure that all the involved interfaces are selected in the **Interfaces** list.

DISCRETIZATION

Select the order of the **Displacement field** — **Linear** or **Quadratic**. The degrees of freedom for the displacement of the shell normals will always have the same shape functions as the displacements.

DEPENDENT VARIABLES

Both interfaces define two dependent variables (fields) — the displacement field **u** and the field of normal displacements **ar**. The names can be changed, but the names of fields and dependent variables must in general be unique within a model. If you intentionally use the same name for fields from different physics interfaces, these degrees of freedom are treated as being the same. This can be used when mixing different type of structural mechanics interfaces, where you often want the displacements to be the equal.

- Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces
- Theory for the Shell and Plate Interfaces

- *Vibrations of a Disk Backed by an Air-Filled Cylinder*: Application Library path **Structural_Mechanics_Module/Acoustic-Structure_Interaction/coupled_vibrations_manual**
- *Pinched Hemispherical Shell*: Application Library path **Structural_Mechanics_Module/Verification_Examples/pinched_hemispherical_shell**

Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces

The Shell and Plate Interfaces have the following domain, boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Shell or Plate interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Boundary** (Shell interface) or **Domain** (Plate interface) level are

- **Material Models**
- **Face and Volume Loads**
- **Mass, Spring, and Damper**
- **Connections**
- **Face Constraints**

The submenus at the **Edge** (Shell interface) or **Boundary** (Plate interface) level are

- **Mass, Spring, and Damper**
- **Connections** (Shell interface only)
- **More Constraints**
- **Pairs**

There is also the **Points** submenu.

LINKS TO FEATURE NODE INFORMATION

These nodes are described in this section (listed in alphabetical order):

- Added Mass
- Adiabatic Heating, Layered
- Antisymmetry
- Attachment
- Body Load
- Boundary to Boundary
- Coupling Loss
- Damping
- Dielectric Loss
- Edge Load
- Edge to Boundary
- External Stress
- Fasteners
- Fiber
- Face Load
- Fixed Constraint
- Hole Selection
- Hygroscopic Swelling
- Inelastic Strain Rate
- Initial Values
- Initial Stress and Strain
- Linear Elastic Material, Layered
- Hyperelastic Material, Layered
- Linear Elastic Material
- Piezoelectric Material, Layered
- Mechanical Damping
- No Rotation
- Periodic Condition
- Phase
- Pinned
- Point Load
- Point Load, Free
- Point Mass
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Rigid Connector
- Ring Load
- Ring Load, Free
- Safety
- Section Stiffness
- Shell Local System
- Simply Supported
- Spot Welds
- Spring Foundation
- Symmetry
- Symmetry Plane
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Attachment)
- Thermal Expansion (Rigid Connector)
- Thermal Expansion (Fiber)
- Thickness and Offset

If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence. The exception is that “Pinned” and “No Rotation” boundary conditions do not override each other.

These nodes are described in the documentation for the Solid Mechanics interface:

- Adhesion
- Applied Force (Rigid Connector)
- Applied Moment (Rigid Connector)
- Base Excitation
- Center of Rotation Nodes
- Component Definition
- Contact
- Creep
- Damage
- Fixed Joint
- Free
- Friction
- Gravity
- Inertia Relief
- Linearly Accelerated Frame[1]
- Mass and Moment of Inertia (Rigid Connector)
- Plasticity
- Point Load (on Axis)
- Reduced Flexible Components
- Rigid Material
- Rigid Motion Suppression
- Rotating Frame[1]
- Safety (for Fasteners)
- Slip Velocity
- Spring–Damper
- Wear
- Viscoelasticity
- Viscoplasticity

[1] This is selected from the **Face and Volume Loads** submenu for this interface.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics Reference Manual*

See Common Physics Interface and Feature Settings and Nodes and Common Feature Nodes in the *COMSOL Multiphysics Reference Manual*. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Initial Values

The **Initial Values** node adds initial values for the translational displacement and velocity field as well as the for the normal displacement and velocity field. It can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. In addition to the default **Initial Values** node always present in the interface, you can add more **Initial Values** nodes if needed.

INITIAL VALUES

Enter values or expressions for the initial values:

- **Displacement field** $\mathbf{u}$ (the displacement components u, v, and w)
- **Velocity field** $\frac{\partial u}{\partial t}$
- **Displacement of shell normals** $\mathbf{ar}$
- **Displacement of shell normals, first time derivative** $\frac{\partial \mathbf{ar}}{\partial t}$

Context Menus

Shell > **Initial Values**

Shell > **Boundaries** > **Initial Values**

Plate > **Initial Values**

Plate > **Domains** > **Initial Values**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Shell** > **Initial Values**

Physics tab with **Plate** selected:

Domains > **Plate** > **Initial Values**

Thickness and Offset

Use the **Thickness and Offset** node to define the thickness for the different parts of a shell or membrane structure. In addition to the default **Thickness and Offset** node always present in the interface, you can add more **Thickness and Offset** nodes if needed.

If the actual midsurface is not on the selected boundaries, you can also prescribe an offset in the direction of the surface normal. The offset is defined as positive if the midsurface is displaced from the meshed boundary in the direction of the positive boundary normal.

THICKNESS AND OFFSET

Enter a value for the **Thickness** d_0 of the selected boundaries. The default is 0.01 m in the Shell and Plate interfaces, and 0.0001 m in the Membrane interface. If an expression is used, the thickness can be variable so that a tapered shell or membrane can be modeled.

You can also add a Thickness Change subnode to take into account that the thickness may change over time.

For the offset, select a **Position** — **Top surface on boundary**, **Midsurface on boundary**, **Bottom surface on boundary**, or **User defined**.

For **User defined**, enter a value or expression in the $z_{\text{rel_offset}}$ field for the offset. It is given as the ratio between the offset distance and half the thickness. A value of +1 means that the actual bottom surface is located on the meshed boundary, and a value of −1 means that the top surface is located on the meshed boundary.

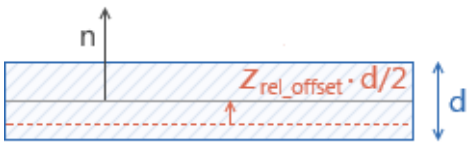


Figure 6-1: Meshed boundary indicated in red. The vector **n** *indicates the positive orientation of the boundary normal.*

Values of $z_{\text{rel_offset}}$ outside the range [-1,1] are also allowed.

The offset settings are not available for the Plate interface.

Thickness and offset settings specified on this node are not applicable for a **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** or **Piezoelectric Material, Layered** node as this information is then provided on **Layered Material** and **Layered Material Link** nodes.

For theory, see Offset.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Thickness and Offset**

Plate > **Thickness and Offset**

Membrane > **Thickness and Offset**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Shell** > **Thickness and Offset**

Physics tab with **Plate** selected:

Domains > **Plate** > **Thickness and Offset**

Physics tab with **Membrane** selected:

Boundaries > **Membrane** > **Thickness and Offset**

Thickness Change

Add a **Thickness Change** node to incorporate an expression for how the thickness of a shell or membrane changes as a function of time. This feature is only used in time-dependent studies. The intended applications are when the thickness varies due to wear, corrosion, electrodeposition, or similar processes. In many cases, the expression for the rate of change will thus depend on variables computed elsewhere.

THICKNESS CHANGE

To incorporate a changing thickness, select **Thickness rate, top surface** and/or **Thickness rate, bottom surface**, as applicable. Then, enter expressions for the thickness change rates, $\partial\Delta_{\text{top}}/\partial t$ and $\partial\Delta_{\text{bot}}/\partial t$.

The current thickness will be computed as $d_0 + \Delta_{\text{top}}(t) + \Delta_{\text{bot}}(t)$, where d_0 is the original thickness given in the parent Thickness and Offset node.

DISCRETIZATION

The additional thickness is a field as function of the spatial coordinates, and it will be approximated by a set of shape functions. Select a **Shape function type** — **From physics**, **Discontinuous Lagrange**, or **Gauss-point data**. When selecting **From physics**, the additional thickness field is approximated by the same shape functions that are used for the dependent variables of the physics.

The order of the shape functions is in all cases selected automatically, based on the discretization in the physics interface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Thickness and Offset** > **Thickness Change**

Membrane > **Thickness and Offset** > **Thickness Change**

Ribbon

Physics tab with **Thickness and Offset** node selected in the model tree:

Attributes > **Thickness Change**

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic shell and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Damping
- Fiber
- Safety

A Shell Local System subnode is always added. In this node you specify the coordinate system in which material orientations and results are interpreted. You can add several **Shell Local System** nodes in order to control the local directions on different boundaries.

LINEAR ELASTIC MATERIAL

Select **Material symmetry**— **Isotropic**, **Orthotropic**, or **Anisotropic** and enter the settings as described for the Linear Elastic Material for the Solid Mechanics interface. Note that:

- For **Orthotropic** no values for E_z, ν_{yz}, or ν_{xz} need to be entered due to the shell assumptions. It is also possible to define **Transversely isotropic** material properties.
- For **User defined Anisotropic** a 6-by-6 symmetric matrix is displayed. Due to the shell assumptions, you only need to enter values for D_{11}, D_{12}, D_{22}, D_{14}, D_{24}, D_{55}, D_{66}, and D_{56}.
- The material orientation is always interpreted in a local coordinate system aligned with the shell boundary as described in Local Coordinate Systems.

OUT-OF-PLANE STRAIN

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If the **Solve for out-of-plane strain components** checkbox is selected, extra degrees of freedom will be added for computing the out-of-plane strain components. This formulation is similar to what is used for plane stress in the Solid Mechanics and Membrane interfaces, and it is computationally somewhat more expensive than the standard formulation. In the default formulation, the out-of-plane strain in the shell is explicitly computed from the stress. This may cause circular references of variables if you for example want the constitutive law to be strain dependent. If you encounter such problems, use the alternative formulation.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Material Models** > **Linear Elastic Material**
Plate > **Material Models** > **Linear Elastic Material**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Material Models** > **Linear Elastic Material**

Physics tab with **Plate** selected:

Domains > **Material Models** > **Linear Elastic Material**

Linear Elastic Material, Layered

The **Linear Elastic Material, Layered** node adds the equations for a layered linear elastic shell.

If the Composite Materials Module analysis is available, this material model can be applied to arbitrary layers in a multilayered shell. The material properties, orientations, and layer thicknesses are defined using **Layered Material** node. The offset, and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

Without the Composite Materials Module, only single layer shells can be modeled. This is still useful. In particular, it is used for nonlinear material models, but also for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered Materials in the documentation for the Composite Materials Module.
- See also the discussion in Layered and Nonlayered Shells.

By adding the following subnodes to the **Linear Elastic Material, Layered** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Damage
- Safety

Some of these material models are only available together with the Nonlinear Structural Materials Module (see www.comsol.com/products/specifications/).

Linear Elastic Material, Layered is only available for the Shell interface, but not for the Plate interface.

SHELL PROPERTIES

For this node, the **Shell Properties** section is mainly used for selecting a material model, but not individual layers. There is one exception: it is possible to select a single stack member. This is useful when combining the Shell interface and the Layered Shell interface on the same boundary (sometimes called the *multiple model method*)

TABLE 6-1: LAYER SELECTIONS; LINEAR ELASTIC MATERIAL, LAYERED.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added	When a Layered Material Stack is used, it is possible to select a single stack member

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

The boundary selection in this node is similar to the **Linear Elastic Material** node. It is however only possible to select boundaries which are part of the selection of a layered material defined in a **Single Layer Material**, **Layered Material Link** or a **Layered Material Stack** node.

LINEAR ELASTIC MATERIAL

Select **Material symmetry**— **Isotropic**, **Orthotropic**, or **Anisotropic** and enter the settings as described for the Linear Elastic Material for the Solid Mechanics interface. If the layers have different types of anisotropy properties, select the one that is most complex.

Note that:

- For **Orthotropic** no values for E_z, ν_{yz}, or ν_{xz} need to be entered due to the shell assumptions. It is also possible to define **Transversely isotropic** material properties.

- For **User defined Anisotropic** a 6-by-6 symmetric matrix is displayed. Due to the shell assumptions, you only need to enter values for $D_{11}, D_{12}, D_{22}, D_{14}, D_{24}, D_{55}, D_{66}$, and D_{56}.
- The material orientation is always interpreted in a laminate coordinate system aligned with the shell boundary as described in Local Coordinate Systems together with the orientation of each layer specified on a layered material.

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which adds an extra dependent variable for either the pressure or for the volumetric strain, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, **Strain formulation**, or **Implicit formulation**.

OUT-OF-PLANE STRAIN

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If the **Solve for out-of-plane strain components** checkbox is selected, extra degrees of freedom will be added for computing the out-of-plane strain components. This formulation is similar to what is used for plane stress in the Solid Mechanics and Membrane interfaces, and it is computationally somewhat more expensive than the standard formulation. See Plane Stress.

In the default formulation, the out-of-plane strain in the shell is explicitly computed from the stress, and no extra degree of freedom is added. This may cause circular references of variables if you for example want the constitutive law to be strain dependent. If you encounter such problems, select the **Solve for out-of-plane strain components** checkbox.

When the Mixed Formulation is used, the **Solve for out-of-plane strain components** checkbox is selected, the extra degrees of freedom are added, and the section **Out-of-Plane Strain** is hidden.

SHEAR CORRECTION FACTOR

In this section there is a list for defining the value of shear correction factors. The two options available are **Automatic** and **User defined**. Once **User defined** option is selected, you can enter the values of k_{23} and k_{13}.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.

- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

The section is available when you also have the Nonlinear Structural Materials Module. Then, to display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

DISCRETIZATION

If **Pressure formulation** is used, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**. If **Strain formulation** is used, select the discretization for the **Auxiliary volumetric strain** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

The **Discretization** section is available when **Pressure formulation** or **Strain formulation** is selected from the **Use mixed formulation** list. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Material Models** > **Linear Elastic Material, Layered**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Material Models** > **Linear Elastic Material, Layered**

Shell Local System

The **Shell Local System** subnode is used for interpreting directions of orthotropic and anisotropic material data as well as when stresses or strains are presented in a local system.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any valid additional coordinate system that the model includes. If the selected coordinate system is not a boundary system, it is projected onto the shell surface as described in Local Coordinate Systems.

When a Shell or Plate interface is added, there is also a **Shell Local System** node added under **Definitions**. This coordinate system can be used to reference the local directions selected in a Shell interface. It is used internally in the Shell interface, and can also be accessed from other physics interfaces. The coordinate system can also be used in a **Coordinate System Surface** plot to visualize the local directions.

- Do not edit or duplicate the **Shell Local System** node added under **Definitions**.
- If you add more than one Shell physics interface to the same component, one **Shell Local System** node is added under **Definitions** by each interface. Do not reference a coordinate system that belongs to another Shell interface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Shell Local System**
Shell > **Rigid Domain** > **Shell Local System**
Plate > **Linear Elastic Material** > **Shell Local System**

Ribbon

Physics tab with **Linear Elastic Material** or **Rigid Domain** node selected in the model tree:

Attributes > **Shell Local System**

Hyperelastic Material, Layered

The **Hyperelastic Material, Layered** node adds the equations for a layered hyperelasticity at large strains. Hyperelastic materials can be suitable for modeling rubber and other polymers, biological tissue, and also for applications in acoustoelasticity.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, this material model can be applied to arbitrary layers in a multilayered shell. The material properties and layer thicknesses are defined using **Layered Material** node. The offset, and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

When a hyperelastic material is included in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** checkbox in the study settings is selected and cannot be cleared.

By adding the following subnodes to the **Hyperelastic Material, Layered** node you can incorporate many other effects.

- Thermal Expansion (for Materials)

- Hygroscopic Swelling
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Polymer Viscoplasticity
- Mullins Effect

Without the Composite Materials Module, only single layer hyperelastic shells can be modeled. This is still useful. In particular, it is used for nonlinear material models, but also for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered and Nonlayered Shells.

Hyperelastic Material, Layered is only available for the Shell interface, but not for the Plate interface.

SHELL PROPERTIES

For this node, the **Shell Properties** section is only used for selecting a material model, but not individual layers.

TABLE 6-2: LAYER SELECTIONS; HYPERELASTIC MATERIAL, LAYERED.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When **Use all layers** is not selected.

Data given in the other sections of this node applies to all layers. Thus, if you enter material data explicitly, rather relying on the default **From material** option, you will override that material property for all selected layers.

- All other settings are described in the documentation for Hyperelastic Material in the Solid Mechanics interface.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate.

BOUNDARY SELECTION

The boundary selection in this node is similar to the **Linear Elastic Material** node. It is however only possible to select boundaries that are part of the selection of a layered material defined in a **Single Layer Material**, **Layered Material Link**, or **Layered Material Stack** node.

HYPERELASTIC MATERIAL

Select a **Material model** and enter the settings as described for the Hyperelastic Material for the Solid Mechanics interface.

SHEAR CORRECTION FACTOR

Enter the shear correction factors for transverse shear k_{23} and k_{13}.

When computing the contribution from shear stiffness to the total virtual work, it is necessary to take into account that the shell approximation assumes that shear stresses and strains in the thickness direction are constant, whereas in reality the distribution is more complicated. The shear correction factors are used to compensate for this, so that the total strain energy density is correct.

ENERGY DISSIPATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Material Models** > **Hyperelastic Material, Layered**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Material Models** > **Hyperelastic Material, Layered**

Piezoelectric Material, Layered

The **Piezoelectric Material, Layered** node defines the piezoelectric material properties either in stress–charge form using the elasticity matrix and the coupling matrix, or in strain–charge form using the compliance matrix and the coupling matrix. It is normally used together with a **Piezoelectricity, Layered** multiphysics coupling node and a corresponding **Piezoelectric Layer** node in the **Electric Currents in Layered Shells** interface.

The **Piezoelectric Material, Layered** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

When working with layered shells, you almost invariably take the material data from what has been defined using **Layered Material Link**, **Layered Material Stack**, or **Single Layer Material** nodes. It is, however, possible to override some data from a **Piezoelectric Material, Layered** node.

In order to have a correct model, all layers must have been assigned material data for all boundaries selected in the settings for the interface. The override rules for the material models in the Shell interface cannot enforce this, in the same way as for other physics interfaces. You cannot have several **Piezoelectric Material, Layered** nodes with the same (or partially overlapping) geometrical selections, but with different layer selections.

By adding the following subnodes to the **Piezoelectric Material, Layered** node you can incorporate many other effects:

- Initial Stress and Strain

- Thermal Expansion (for Materials)
- Mechanical Damping
- Coupling Loss
- Dielectric Loss

When the **Piezoelectric Material, Layered** node is added to the Shell interface in the absence of an active **Piezoelectricity, Layered** multiphysics coupling node, the material behaves similarly to a Linear Elastic Material, Layered node. The elastic properties will correspond to the elasticity or compliance matrix entered (see below). The piezoelectric effect is then *not* included in the equation system.

See also Piezoelectricity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

For this node, the **Shell Properties** section is only used for selecting a material model, but not individual layers.

TABLE 6-3: LAYER SELECTIONS; HYPERELASTIC MATERIAL, LAYERED.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When **Use all layers** is not selected.

Data given in the other sections of this node applies to all layers. Thus, if you enter material data explicitly, rather than relying on the default **From material** option, you will override that material property for all selected layers.

BOUNDARY SELECTION

The boundary selection in this node is similar to the **Linear Elastic Material** node. It is, however, only possible to select boundaries which are part of the selection of a layered material defined in a **Single Layer Material**, **Layered Material Link**, or **Layered Material Stack** node.

PIEZOELECTRIC MATERIAL PROPERTIES

Select a **Constitutive relation** — **Stress–charge form** or **Strain–charge form**. For each of the following, the default uses values **From material**. For **User defined** enter other values in the matrix or field as needed.

- For **Stress–charge form**, select an **Elasticity matrix, Voigt notation** (c_{E}).
- For a **Strain–charge form**, select a **Compliance matrix, Voigt notation** (s_{E}).
- Select a **Coupling matrix, Voigt notation** (d).
- Select a **Relative permittivity** (e_{rS} or e_{rT}).
- Enter values for the **Remanent electric displacement** D_{r}.
- Select a **Density** (p).

For entering these matrices, use the following order (Voigt notation), which is the common convention for piezoelectric materials: xx, yy, zz, yz, xz, zy.

Density

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

The density is for dynamic analysis, and also when computing mass forces for gravitational or rotating frame loads.

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

OUT-OF-PLANE MATERIAL ORIENTATION

The layered material always operates with a boundary coordinate system on the base surface (laminate system). For such systems, the third base vector direction is always

normal to the surface. Use a special control available in this section if you need to change the out-of-plane orientation of the material. This is essential if your piezoelectric device requires the pole direction to be tangential to the shell, but the pole direction in the material data coincides with the third coordinate axis — such material orientation is assumed for all the piezoelectric material data available in COMSOL Material Library.

SHEAR CORRECTION FACTOR

In this section there is a list for defining the value of shear correction factors. The two options available are **Automatic** and **User defined**. Once **User defined** option is selected, you can enter the values of k_{23} and k_{13}.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by **Mechanical damping**.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Material Models** > **Piezoelectric Material, Layered**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Material Models** > **Piezoelectric Material, Layered**

Viscoelasticity

Use the **Viscoelasticity** subnode to add viscous stress contributions to a Linear Elastic Material, Layered or to a Hyperelastic Material, Layered.

If the Composite Materials Module analysis is available, the viscoelastic model can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this viscoelastic model is to be used.

TABLE 6-4: LAYER SELECTIONS; VISCOELASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the viscoelastic model differs between layers, you will need to add several **Viscoelasticity** nodes with different layer selections.

If the same layer is selected in two **Viscoelasticity** nodes being active on the boundary, then there will be two viscoelastic contributions.

- All other settings are described in the documentation for Viscoelasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Viscoelasticity**
Shell > **Hyperelastic Material, Layered** > **Viscoelasticity**

Ribbon

Physics tab with **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Viscoelasticity**

Plasticity

Use the **Plasticity** subnode to define the properties for modeling elastoplastic materials. This material model can be used together with the Linear Elastic Material, Layered.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the plasticity model can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this plasticity model is to be used.

TABLE 6-5: LAYER SELECTIONS; PLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the plasticity model differs between layers, you will need to add several **Plasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Plasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If there are two **Plasticity** nodes where the same layer is selected on the same boundary, the latter will be overridden on the common selections.

- All other settings are described in the documentation for Plasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Plasticity**
Shell > **Hyperelastic Material, Layered** > **Plasticity**

Ribbon

Physics tab with **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Plasticity**

Set Variables

Use the **Set Variables** subnode to Plasticity of a Linear Elastic Material, Layered or Hyperelastic Material, Layered to reset plasticity variables according to a **Setting condition** that you define. When the **Setting condition** is satisfied, the plasticity variables are reset to the specified values.

SET VARIABLES

Enter the **Setting condition**. This is a Boolean expression that will determine when the plastic variables are reset.

From the **Equivalent plastic strain** list, select **Do not set** or **User defined**. The default **User defined** value is zero. Depending on the type of plasticity model, set additional plasticity variables.

From the **Plastic strain tensor** list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components of the **Plastic strain tensor**.

If **Armstrong–Frederick** or **Chaboche** is selected from the Kinematic Hardening list, specify the values for the components of the **Back strain tensor**. From the **Back strain tensor** list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components of the **Back strain tensor**.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Plasticity** > **Set Variables**
Shell > **Hyperelastic Material, Layered** > **Plasticity** > **Set Variables**

Ribbon

Physics tab with **Plasticity** node selected in the model tree:

Attributes > **Set Variables**

Creep

Use the **Creep** subnode to define the creep properties of a Linear Elastic Material, Layered or a Hyperelastic Material, Layered node.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, creep models can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this creep model is to be used.

TABLE 6-6: LAYER SELECTIONS; CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the creep model differs between layers, you will need to add several **Creep** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Creep**

Shell > **Hyperelastic Material, Layered** > **Creep**

Ribbon

Physics tab with **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Creep**

Additional Creep

Use the **Additional Creep** subnode to define additional contributions to the creep model defined by the parent Creep node, such as primary or tertiary creep behavior. A **Creep** node can have any number of **Additional Creep** subnodes with different settings to model advanced creep mechanisms.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the creep model can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which the additional creep model is to be used.

TABLE 6-7: LAYER SELECTIONS; ADDITIONAL CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the creep model differs between layers, you will need to add several **Additional Creep** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Additional Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Additional Creep** nodes being active on the boundary, then there will be two creep strain contributions

- All other settings are described in the documentation for Additional Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Creep** > **Additional Creep**

Shell > **Hyperelastic Material, Layered** > **Creep** > **Additional Creep**

Ribbon

Physics tab with **Creep** node selected in the model tree:

Attributes > **Additional Creep**

Viscoplasticity

Use the **Viscoplasticity** subnode to define the viscoplastic properties of a Linear Elastic Material, Layered or a Hyperelastic Material, Layered node.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the viscoplastic model can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this viscoplasticity model is to be used.

TABLE 6-8: LAYER SELECTIONS; VISCOPLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the viscoplasticity model differs between layers, you will need to add several **Viscoplasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Viscoplasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Viscoplasticity** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Viscoplasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Viscoplasticity**
Shell > **Hyperelastic Material, Layered** > **Viscoplasticity**

Ribbon

Physics tab with **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Viscoplasticity**

Polymer Viscoplasticity

Use the **Polymer Viscoplasticity** subnode to define the viscoplastic properties of a Hyperelastic Material, Layered node.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, viscoplastic models can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this viscoplastic model is to be used.

TABLE 6-9: LAYER SELECTIONS; POLYMER VISCOPLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the viscoplastic model differs between layers, you will need to add several **Polymer Viscoplasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Polymer Viscoplasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Polymer Viscoplasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Hyperelastic Material, Layered** > **Polymer Viscoplasticity**

Ribbon

Physics tab with **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Polymer Viscoplasticity**

Additional Network

Use the **Additional Network** subnode to define additional contributions to the Parallel Network model defined by the parent Polymer Viscoplasticity node. A **Polymer Viscoplasticity** node can have any number of **Additional Network** subnodes with different settings to model advanced viscoplastic mechanisms.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the creep model can be applied to arbitrary layers in a multilayered shell.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

Select the layer or layers for which this creep model is to be used.

TABLE 6-10: LAYER SELECTIONS; ADDITIONAL NETWORK.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the *Parallel Network* model differs between layers, you will need to add several **Additional Networks** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Additional Network** node where

From material is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Additional Network** nodes being active on the boundary, then there will be two additional contributions

- All other settings are described in the documentation for Additional Network in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Hyperelastic Material, Layered** > **Polymer Viscoplasticity** > **Additional Network**

Ribbon

Physics tab with **Polymer Viscoplasticity** node selected in the model tree:

Attributes > **Additional Network**

Fiber

Add one or more **Fiber** nodes to add stiffness in specified directions. The **Fiber** feature can be used together with a **Linear Elastic Material**. The assumption is that the volume fraction of the fibers is small when compared to the shell volume.

You can include the effect of thermal expansion of the fiber by adding a **Thermal Expansion** subnode (see Thermal Expansion (Fiber)).

FIBER MODEL

Select an **Approximation** — **Wire** or **Beam**. In the wire approximation, only the axial stress in the fiber is taken into account. When using the beam approximation, there will also be a contribution to the bending stiffness, based on the diameter of the fibers. In either case, the fiber will contribute to the bending stiffness of the shell as long as it is not placed at the midsurface.

Select a **Material Model** — **Linear elastic** or **Uniaxial data**. The uniaxial data model is only available when the wire approximation is used.

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material.

- For **Linear elastic** enter the **Young's modulus** E_{fiber} and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions.
- For **Uniaxial data** enter the **Uniaxial stress function** $\sigma_{\text{ax,a}}$ and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions. The default user defined expression for σ_{a} is the linear function `210[GPa]*<item>.emela`, where the variable `<item>.emela` is the elastic strain in the fiber direction, and `<item>` corresponds to the tag of the **Fiber** node, for example `shell.lemm1.fib1`. See Fibers for Linear and Nonlinear Elastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material.

DISTRIBUTION AND ORIENTATION

From the **Specify** list select how the fiber volume fraction is entered — **Volume distribution** or **Surface distribution**.

- For **Volume distribution**, enter the fiber **Volume fraction** v_{fiber}. To be consistent with the underlying assumptions, it should not exceed a few percent.
- For **Surface distribution**, enter the fiber **Fiber placement**.

  For a **Linear elastic** material, the **Fiber placement** can be at — **Top surface**, **Midsurface**, **Bottom surface**, or **User defined**. Fibers represented by a **Uniaxial data** material model are always assumed to be placed at the shell midsurface.

  For **User defined**, enter a value or expression for the relative offset $z_{\text{fiber_rel}}$. It is given as the ratio between the offset distance and half the shell thickness. A value of +1 means that the center if the fiber is located on the top surface of the shell, and a value of –1 means that the center of the fiber is located on the bottom surface.

  Enter the fiber **Diameter** d_{fiber} and the fiber **Spacing** s_{fiber} to compute the fiber volume fraction. To be consistent with the underlying assumptions, the diameter–spacing ratio should not exceed a few percent. When a beam assumption is used, the fiber diameter is used to compute the bending stiffness.

Select the **Fiber orientation** from the list. The available choices are the axis directions of the coordinate system selected in the **Coordinate System Selection** section, or **User defined**. For **User defined**, enter a **Direction**. The direction vector $\mathbf{a}$ is interpreted in the in-plane directions of the selected local coordinate system.

See also Distributed Fiber Models in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Fiber**

Plate > **Linear Elastic Material** > **Fiber**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Fiber**

Thermal Expansion (Fiber)

It is possible to model thermal expansion in the fibers when these have different thermal properties than the surrounding matrix. The assumption is that the volume of fibers is small when compared to the base material, and that thermal expansion (or contraction) occurs in the fiber direction only.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains in the fibers. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

The fiber **Temperature** T is by default obtained from a **Common model input**. You can also select an existing temperature variable from a heat transfer interface, if any temperature variables exist, or manually enter a value or expression by selecting **User defined**.

THERMAL EXPANSION PROPERTIES

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's thermal expansion properties. In most cases, you would use the

boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material.

Select an **Input type** to specify how the thermal strains in the fibers should be modeled. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain in the fibers is given by

$$\varepsilon_{\text{th,fib}} = \alpha(T - T_{\text{ref}})$$

Here, the secant coefficient of thermal expansion α can be temperature-dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain in the fibers is given by

$$\varepsilon_{\text{th,fib}} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter explicitly the thermal strain in the fibers dL as function of temperature.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, enter the coefficient of the thermal expansion or the thermal strain in the fiber direction $\mathbf{a}$, since thermal expansion is considered in the fiber direction only.

In all cases, the contribution to the thermal strain tensor is defined from the thermal strain in the fibers $\varepsilon_{\text{th,fib}}$, and the fiber direction $\mathbf{a}$ as

$$\varepsilon_{\text{th}} = \varepsilon_{\text{th,fib}}\mathbf{a} \otimes \mathbf{a}$$

THERMAL BENDING

When **Approximation** in the parent Fiber node is set to **Beam**, or when it is set to **Wire** and the **Fiber placement** is set to **Volume distribution**, you can take local thermal bending of the fiber into account.

From the list, select **Temperature difference in thickness direction** or **Temperature gradient in thickness direction**. In either case, the values given refer to the thickness of the base material. Typically, you would thus enter the same data as in the **Thermal Expansion** node of the base material.

When **Temperature difference in thickness direction** is selected, enter the temperature difference ΔT_z. This is the temperature difference between the top and bottom surfaces.

When **Temperature gradient in thickness direction** is selected, enter the temperature gradient T' in the direction from the bottom surface to the top surface.

See also Thermal Expansion in Fibers in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**

Plate > **Linear Elastic Material** > **Fiber** > **Thermal Expansion**

Ribbon

Physics tab with **Fiber** node selected in the model tree:

Attributes > **Thermal Expansion**

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature. It is possible to model bending due to a temperature gradient in the thickness direction of the shell.

Thermal expansion can be modeled for the Linear Elastic Material, Linear Elastic Material, Layered, Hyperelastic Material, Layered, or Piezoelectric Material, Layered. The thermal expansion can be applied to arbitrary layers in a multilayered shell when the Composite Materials Module analysis is available.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

This section is present when this node is added under a **Linear Elastic Material, Layered** or **Layered Hyperelastic Elastic Material** node. In this section, select the layers in which thermal expansion needs to be modeled.

TABLE 6-11: LAYER SELECTIONS; THERMAL EXPANSION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

For a multilayered shell, it is often easiest to add one **Thermal Expansion** node per layer, if the temperature input is manual.

If the same layer is selected in two **Thermal Expansion** nodes being active on the same boundary, the second definition will override the previous.

- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the expansion with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

The **Temperature** T is by default obtained from a **Common model input**. You can also select an existing temperature variable from a heat transfer interface (for example, **Temperature (htsh/sol1)**), if any temperature variables exist, or manually enter a value or expression by selecting **User defined**. This is the midsurface temperature of the shell, controlling the membrane part of the thermal expansion. For layered shells, it is the mid-layer temperature for each layer.

If needed, you can add a through-thickness temperature gradient in the **Thermal Bending** section.

Thermal strains are proportional to the temperature, while structural strains are computed from the gradient of the displacement field. It is good practice to match the discretization order of thermal and structural strains.

When adding a **Thermal Expansion** subnode, and the temperature field is computed by another physics interface (often the Heat Transfer in Shells interface); use a discretization one order lower for the temperature field than what is used for the displacement field.

See also:

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*

THERMAL EXPANSION PROPERTIES

Select an **Input type** to specify how the thermal strain is entered. For the default **Secant coefficient of thermal expansion** the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

where the secant coefficient of thermal expansion α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components

for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When **Diagonal** or **Symmetric** input is used, the axis orientations are given by the coordinate system selection in the parent node.

THERMAL BENDING

The settings in this section differ slightly depending on if the **Thermal Expansion** subnode is added under **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material, Layered**, **Piezoelectric Material, Layered**, or **Section Stiffness**.

If the temperature distribution is obtained from another physics interface of a layered type, such as Heat Transfer in Shells, then the temperature variation in the through-thickness direction is automatically known. No input in this section is needed.

The temperature is assumed to vary linearly through the thickness.

From the list, select **Temperature difference in thickness direction** or **Temperature gradient in thickness direction**.

Linear Elastic Material or Section Stiffness

When **Temperature difference in thickness direction** is selected, enter the temperature difference ΔT_z. This is the temperature difference between the top and bottom surfaces.

When **Temperature gradient in thickness direction** is selected, enter the temperature gradient T' in the direction from the bottom surface to the top surface.

Linear Elastic Material, Layered, Hyperelastic Material, Layered Or Piezoelectric Material, Layered

When **Temperature difference in thickness direction** is selected, enter the temperature difference ΔT_z between the top surface of the topmost of the selected layers and bottom surface of the bottommost of the selected layers.

When **Temperature gradient in thickness direction** is selected, enter the temperature gradient T' in the direction from the bottom surface to the top surface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Thermal Expansion**

Shell > **Linear Elastic Material, Layered** > **Thermal Expansion**
Shell > **Hyperelastic Material, Layered** > **Thermal Expansion**
Shell > **Piezoelectric Material, Layered** > **Thermal Expansion**
Shell > **Section Stiffness** > **Thermal Expansion**
Plate > **Linear Elastic Material** > **Thermal Expansion**
Plate > **Section Stiffness** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material, Layered**, **Piezoelectric Material, Layered**, or **Section Stiffness** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the thickness direction of the shell.

Hygroscopic swelling can be modeled for the Linear Elastic Material, Linear Elastic Material, Layered, and Hyperelastic Material, Layered. The swelling can be applied to arbitrary layers in a multilayered shell when the Composite Materials Module analysis is available.

See also Layered and Nonlayered Shells.

SHELL PROPERTIES

This section is present when this node is added under a **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node. In this section, select the layers in which hygroscopic swelling needs to be modeled.

TABLE 6-12: LAYER SELECTIONS; HYGROSCOPIC SWELLING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the **Moisture concentration type** differs between layers, you will need to add several **Hygroscopic Swelling** nodes with different layer selections. If only the material data values differ, you can use a single **Hygroscopic Swelling** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

For a multilayered shell, it is often easiest to add one **Hygroscopic Swelling** node per layer, if the temperature input is manual.

If the same layer is selected in two **Hygroscopic Swelling** nodes being active on the same boundary, the second definition will override the previous.

- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the expansion with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

MODEL INPUTS

From the **Concentration** *c* list, select an existing concentration variable from another physics interface, if any concentration variables exist. For **User defined** enter a value or expression for the concentration. This is the midsurface concentration in the shell, controlling the membrane part of the hygroscopic swelling.

The unit for the input depends on the setting of **Concentration type** in the **Hygroscopic Swelling Properties** section. Only concentration variables having the chosen physical dimension are available in the **Concentration** list.

If needed, you can add a through-thickness concentration gradient in the **Hygroscopic Bending** section.

Hygroscopic strains are proportional to the concentration, while structural strains are computed from the gradient of the displacement field. It is good practice to match the discretization order of hygroscopic and structural strains.

When adding a **Hygroscopic Swelling** subnode, and the concentration field is computed by another physics interface; use a discretization one order lower for the concentration field than what is used for the displacement field.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** (the default) or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref}. This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as **Concentration type**, also enter the **Molar mass** of the fluid, M_{m}. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h}. The default value for the **User defined** case is `1.5e-4` m^3/kg. When a nonisotropic coefficient of hygroscopic swelling is used, the axis orientations are given by the coordinate system selection in the parent node.

The **Include moisture as added mass** checkbox is selected by default. When selected, the mass of the fluid is included in a dynamic analysis, and when using mass proportional loads. It will also contribute when computing mass properties.

HYGROSCOPIC BENDING

The settings in this section differ slightly depending on if the **Hygroscopic Swelling** subnode is added under **Linear Elastic Material**, **Linear Elastic Material, Layered**, or **Hyperelastic Material, Layered**.

The concentration is then assumed to vary linearly through the thickness.

Linear Elastic Material

Enter the **Concentration difference in thickness direction** Δc_z. This is the difference between the concentrations at the top and bottoms surfaces.

Linear Elastic Material, Layered or Hyperelastic Material, Layered

From the list, select **Concentration difference in thickness direction** or **Concentration gradient in thickness direction**.

When **Concentration difference in thickness direction** is selected, enter the concentration difference Δc_z between the top surface of the topmost of the selected layers and bottom surface of the bottommost of the selected layers.

When **Concentration gradient in thickness direction** is selected, enter the concentration gradient c' in the direction from the bottom surface to the top surface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Hygroscopic Swelling**
Shell > **Linear Elastic Material, Layered** > **Hygroscopic Swelling**
Shell > **Hyperelastic Material, Layered** > **Hygroscopic Swelling**
Plate > **Linear Elastic Material** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, or **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to the Linear Elastic Material, Linear Elastic Material, Layered, Piezoelectric Material, Layered or Section Stiffness material models in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense but rather a contribution to the constitutive relation

When used with **Linear Elastic Material, Layered**, or **Piezoelectric Material, Layered** the contribution can be applied to arbitrary layers in a multilayered shell when the Composite Materials Module is available.

See also Layered and Nonlayered Shells.

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** or **Piezoelectric Material, Layered** node. Select the layers in which initial stress and strain needs to be modeled.

TABLE 6-13: LAYER SELECTIONS; INITIAL STRESS AND STRAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the initial stress or strain differ between layers, you will need to add several **Initial Stress and Strain** nodes with different layer selections.

If the same layer is selected in two **Initial Stress and Strain** nodes being active on the same boundary, the contributions will be added.

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

COORDINATE SYSTEM SELECTION

Select a **Local coordinate system** defining the directions along which the initial stresses and strains are given.

As Subnode to Linear Elastic Material

The **Shell local system** is selected by default. This means that the orientations are the same as used for the material description.

If you select **Projected from coordinate system**, you can select any coordinate system present in the model. The **Coordinate system** list will then contain any coordinate system present in the model. The coordinate system given is projected onto the shell surface as described in Local Coordinate Systems.

As Subnode to Linear Elastic Material, Layered or Section Stiffness

You can select any available boundary system.

INITIAL STRESS

Specify the initial stress as the **Initial in-plane force**, the **Initial moment**, and the **Initial out-of-plane shear force**. Enter values or expressions in the applicable fields for the:

- **Initial in-plane force** N_0
- **Initial moment** M_0
- **Initial out-of-plane shear force** Q_0

INITIAL STRAIN

Specify the initial strain as the **Initial membrane strain**, the **Initial bending strain**, and the **Initial transverse shear strain**. Enter values or expressions in the applicable fields for the:

- **Initial membrane strain** γ_0
- **Initial bending strain** χ_0
- **Initial transverse shear strain** ζ_0

- For definitions of the generalized strains, see Theory for the Shell and Plate Interfaces.
- For details about initial stresses and strains, see Inelastic Strain Contributions and Initial Stress and Strain.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Initial Stress and Strain**
Shell > **Linear Elastic Material, Layered** > **Initial Stress and Strain**
Shell > **Piezoelectric Material, Layered** > **Initial Stress and Strain**

Shell > **Section Stiffness** > **Initial Stress and Strain**
Plate > **Linear Elastic Material** > **Initial Stress and Strain**
Plate > **Section Stiffness** > **Initial Stress and Strain**

Ribbon
Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Piezoelectric Material, Layered**, or **Section Stiffness** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode to the Linear Elastic Material, Linear Elastic Material, Layered, Hyperelastic Material, Layered, or Section Stiffness material models in order to specify an additional stress contribution which is not part of the constitutive relation. The external stress can be added to the total stress tensor, or act only as an extra load contribution.

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor, whereas the option to use the stress as only a load contribution exists only in **External Stress**.

SHELL PROPERTIES

This section is present when this node is added under a **Linear Elastic Material, Layered** or **Hyperelastic Material, Layered** node.

Select the layers in which external stress is to be modeled.

TABLE 6-14: LAYER SELECTIONS; EXTERNAL STRESS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the external stress input type differs between layers, you will need to add several **External Stress** nodes with different layer selections.

If the same layer is selected in two **External Stress** nodes being active on the same boundary, the contributions will be added.

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

COORDINATE SYSTEM SELECTION

Select a **Local coordinate system** defining the directions along which the components of the input data are given.

Linear Elastic Material

The **Shell local system** is selected by default. This means that the orientations are the same as used for the material description.

If you select **Projected from coordinate system**, you can select any coordinate system present in the model. The **Coordinate system** list will then contain any coordinate system present in the model. The coordinate system given is projected onto the shell surface as described in Local Coordinate Systems.

Linear Elastic Material, Layered, Hyperelastic Material, Layered, or Section Stiffness

You can select any available boundary system.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor (Material)**, **Stress tensor (Spatial)**, or **Section forces**. This selection is not available when the parent node is Section Stiffness, in which case only the section force type of input is possible.

- When **Stress tensor (Material)** is selected, you enter the external stress in the form of Second Piola–Kirchhoff stress tensors. The **Membrane part of external stress**, **Bending part of external stress**, and **Shear part of external stress** lists all behave the same way: Depending on the type, they will contain all membrane (bending, shear) stress tensors announced by any physics interface, and the **User defined** entry. When **User defined** is selected, you can enter the data for the membrane (bending, shear) part of the **External stress tensor** $S_{\text{ext,m}}$ ($S_{\text{ext,b}}$, $S_{\text{ext,s}}$) as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface

is selected, the coordinate system setting is ignored — the orientation is handled internally.

- When **Stress tensor (Spatial)** is selected, you enter the external stress in the form of Cauchy stress tensors. The **Membrane part of external stress**, **Bending part of external stress**, and **Shear part of external stress** lists all behave the same way: you can enter the data for the membrane (bending, shear) part of the **External stress tensor** $s_{\text{ext,m}}$ ($s_{\text{ext,b}}$, $s_{\text{ext,s}}$) as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system.
- When **Section forces** is selected, specify the external stress as values or expressions for the **In-plane force** N_{ext}, the **Moment** M_{ext}, and the **Out-of-plane shear force** Q_{ext}.

Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** to determine the effect of the contribution.

Selecting a stress tensor announced by the same physics interface as where the **External Stress** node is added, will result in an error ('`Circular variable dependency detected`'). This operation would imply that the computed stress depends on itself.

For theory, see External Stress.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **External Stress**
Shell > **Linear Elastic Material, Layered** > **External Stress**
Shell > **Hyperelastic Material, Layered** > **External Stress**
Shell > **Section Stiffness** > **External Stress**
Plate > **Linear Elastic Material** > **External Stress**
Plate > **Section Stiffness** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Hyperelastic Material, Layered**, or **Section Stiffness** node selected in the model tree:

Attributes > **External Stress**

External Strain

Many of the material models in COMSOL Multiphysics will compute a stress based on an *elastic strain*. The elastic strain tensor is obtained after removing any inelastic deformation contribution from the total deformation from the displacements.

In the finite deformation case, the inelastic strain is instead removed using a multiplicative decomposition of the deformation gradient tensor. The *elastic deformation* gradient tensor is the basis for all strain energy formulations in hyperelastic materials, and also for the elastic strain in linear and nonlinear elasticity. It is derived by removing the inelastic deformation from the total deformation gradient tensor. See Multiplicative Decomposition in the *Structural Mechanics Theory* chapter.

The **External Strain** subnode allows you to provide inelastic strain contributions to the Hyperelastic Material, Layered.

SHELL PROPERTIES

See the documentation for the External Strain node in the Layered Shell chapter.

COORDINATE SYSTEM SELECTION

Select a **Coordinate system**. All inputs that you give below are interpreted in the given coordinate system. Deformation gradients are rotated by the local system in both indices.

EXTERNAL STRAIN

Select the type of **Strain input** — **Deformation gradient**, **Deformation gradient, inverse**, or **Stretches**.

Deformation Gradient

For **Deformation gradient**, enter an inelastic deformation gradient contribution $\boldsymbol{F}_{\text{ext}}$. From the **External deformation gradient** list, you can choose **User defined**, or any deformation gradient tensor which is announced by another physics interface. If you select **User defined**, enter values or expressions for the components of the deformation gradient tensor.

Deformation Gradient, Inverse

For **Deformation gradient, inverse**, enter an inelastic inverse deformation gradient contribution $\boldsymbol{F}_{\text{ext}}^{-1}$. From the **External deformation gradient inverse** list, you can choose **User defined**, or any inverse deformation gradient tensor which is announced by

another physics interface. If you select **User defined**, enter values or expressions for the components of the inverse deformation gradient tensor.

Stretches

For **Stretches**, enter values or expressions for the three principal stretches. Entering data on this form is convenient for some simple geometries and strain states, but in general it is difficult to provide suitable a coordinate system for the principal orientations.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Hyperelastic Material, Layered** > **External Strain**

Ribbon

Physics tab with **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **External Strain**

Inelastic Strain Rate

The **Inelastic Strain Rate** subnode allows you to provide inelastic strain contributions to the material models **Linear Elastic Material, Layered** and **Hyperelastic Material, Layered** on a variety of formats as in External Strain. Here, however, the inelastic contribution is given as a rate, and the total inelastic contribution is computed by integrating this rate in time.

SHELL PROPERTIES

See the documentation for the Inelastic Strain Rate node in the Layered Shell chapter.

All other settings are described in the documentation for Inelastic Strain Rate in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Inelastic Strain Rate**
Shell > **Hyperelastic Material, Layered** > **Inelastic Strain Rate**

Ribbon

Physics tab with **Linear Elastic Material, Layered** node selected in the model tree:

Damage

The **Damage** subnode allows you to model damage and cracking in brittle materials according to various criteria. It is available in the Solid Mechanics, Shell, and Layered Shell interfaces, and it can be used in combination with the Linear Elastic Material, Layered.

SHELL PROPERTIES

Select the layer or layers for which the damage is to be computed.

TABLE 6-15: LAYER SELECTIONS; DAMAGE.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the damage input type differs between layers, you will need to add several **Damage** nodes with different layer selections.

If the same layer is selected in two **Damage** nodes being active on the same boundary, the contributions will be overridden.

- All other settings are described in the documentation for Damage in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

See also

- Modeling Damage in the *Structural Mechanics Modeling* chapter.
- Damage Models in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material, Layered** > **Damage**

Ribbon

Physics tab with **Linear Elastic Material, Layered** node selected in the model tree:

Attributes > **Damage**

Mullins Effect

Some nonlinear effects observed in rubbers, such as hysteresis in stress–stretch curves, residual strains, and stress softening effects, are not accounted in the formulation of common hyperelastic materials. See Mullins Effect in the *Structural Mechanics Theory* chapter.

Use the **Mullins effect** subnode to define the properties for modeling the stress-softening phenomenon under cyclic loading. The Mullins effect can be used together with Hyperelastic Material, Layered.

SHELL PROPERTIES

See the documentation for the Mullins Effect node in the Layered Shell chapter.

MULLINS EFFECT

The **Damage function** defines the model for Mullins effect. Select **Ogden–Roxburgh** or **Miehe**.

For **Ogden–Roxburgh** enter the following settings:

- **Maximum damage** d_∞.The default is 1.
- **Damage saturation** W_{sat}. The default is 1 MJ/m^3.
- **Deformation dependence coefficient** α. The default is 0.

For **Miehe** enter the following settings:

- **Maximum damage** d_∞.The default is 1.

Damage saturation W_{sat}. The default is 1 MJ/m^3.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Hyperelastic Material, Layered** > **Mullins Effect**

Ribbon

Physics tab with **Hyperelastic Material, Layered** node selected in the model tree:

Attributes > **Mullins Effect**

Safety

Use the **Safety** subnode to set up variables which can be used to check the risk of failure according to various criteria. It can be used in combination with **Linear Elastic Material**, and **Linear Elastic Material, Layered**. Four different variables describing the failure risk are defined, as described in Table 6-16.

You can add any number of **Safety** nodes to a single material model. The contents of this feature do not affect the analysis results as such, so you can add **Safety** nodes after having performed an analysis, and just do an **Update Solution** in order to access to the new variables for result evaluation.

TABLE 6-16: VARIABLES FOR SAFETY FACTOR EVALUATION.

VARIABLE	DESCRIPTION	CRITERION FULFILLED	CRITERION VIOLATED
Failure index, FI	For a linear criterion, this is the ratio between the computed value and the given limit.	FI<1	FI>1
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=1
Safety factor, SF	For a linear criterion, this is 1/FI.	SF>1	SF<1
Margin of safety, MoS	SF-1	MoS>0	MoS<0

For orthotropic and anisotropic failure criteria, the directions are given by the coordinate system selection in the parent node.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** node.

Select the layer or layers for which the safety factor variables are to be defined.

TABLE 6-17: LAYER SELECTIONS; SAFETY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Each **Safety** subnode defines its own set of variables, so there is no interaction if you add several such nodes with the same selection.

- All other settings are described in the documentation for Safety in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe parameters for the allowable values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Variables** > **Safety**
Shell > **Linear Elastic Material, Layered** > **Variables** > **Safety**
Plate > **Linear Elastic Material** > **Variables** > **Safety**

Ribbon

Physics tab with **Linear Elastic Material** or **Linear Elastic Material, Layered** node selected in the model builder tree:

Attributes > **Variables** > **Safety**

Damping

Use the **Damping** subnode to add several types of damping to the material model. Damping can be used in Time Dependent, Eigenfrequency, and Frequency Domain studies; for other study types the settings in the **Damping** subnode are ignored.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** node.

Select the layer or layers for which this damping model is to be used.

TABLE 6-18: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Damping** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Damping** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Damping in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe damping values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Linear Elastic Material** > **Damping**
Shell > **Linear Elastic Material, Layered** > **Damping**
Shell > **Layered Linear Hyperelastic Material** > **Damping**
Shell > **Section Stiffness** > **Damping**
Plate > **Linear Elastic Material** > **Damping**
Plate > **Section Stiffness** > **Damping**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Layered Hyperelastic Elastic Material**, or **Section Stiffness** node selected in the model tree:

Attributes > **Damping**

Mechanical Damping

The **Mechanical Damping** subnode allows you to model mechanical losses in the Piezoelectric Material, Layered either using loss factor material data for the stiffness, or in the form of Rayleigh proportional damping.

SHELL PROPERTIES

Select the layer or layers for which this damping model is to be used.

TABLE 6-19: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Mechanical Damping** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Mechanical Damping** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Mechanical Damping in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe damping values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Piezoelectric Material, Layered** > **Mechanical Damping**

Ribbon

Physics tab with **Piezoelectric Material, Layered** node selected in the model tree:

Attributes > **Mechanical Damping**

Coupling Loss

The **Coupling Loss** subnode allows you to model losses in the piezoelectric coupling in a Piezoelectric Material, Layered, either by using the loss factor material data for the coupling matrix or as Rayleigh proportional damping.

SHELL PROPERTIES

Select the layer or layers for which this loss model is to be used.

TABLE 6-20: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Coupling Loss** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Coupling Loss** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Coupling Loss in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe loss values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Piezoelectric Material, Layered** > **Coupling Loss**

Ribbon

Physics tab with **Piezoelectric Material, Layered** node selected in the model tree:

Attributes > **Coupling Loss**

Dielectric Loss

The **Dielectric Loss** subnode allows you to model losses in the electrical losses in a Piezoelectric Material, Layered. The losses can be prescribed either by using a loss factor for the electric permittivity, or in the form of dielectric dispersion.

SHELL PROPERTIES

Select the layer or layers for which this loss model is to be used.

TABLE 6-21: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Dielectric Loss** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Dielectric Loss** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Dielectric Loss in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe loss values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Piezoelectric Material, Layered** > **Dielectric Loss**

Ribbon

Physics tab with **Piezoelectric Material, Layered** node selected in the model tree:

Attributes > **Dielectric Loss**

Section Stiffness

The **Section Stiffness** material model provides a way of directly entering the stiffness or compliance of a shell without direct knowledge about the cross-section thickness and material distribution. Its main purpose is for entering homogenized properties of for example perforated or corrugated sheets.

Since only the stiffness, but not the section geometry is known, it is not possible to deduce the stresses. You can however provide an expression for stress computation based on the computed section forces.

In many cases, the properties of the shell cross section are anisotropic. All input data are interpreted along the local directions specified in the **Coordinate System Selection** section.

By adding the following subnodes to the **Section Stiffness** node you can incorporate other effects:

- Thermal Expansion (for Materials)
- Initial Stress and Strain
- External Stress
- Damping

SECTION PROPERTIES

From the **Specify** list, select **Effective stiffness** or **Effective flexibility** in order to specify the representation of the section stiffness.

For **Effective stiffness**, enter the **Extensional stiffness matrix**, $\mathbf{D}_A$; **Bending–extensional stiffness matrix**, $\mathbf{D}_B$; **Bending stiffness matrix**, $\mathbf{D}_D$; and **Shear stiffness matrix**, $\mathbf{D}_{As}$.

For **Effective flexibility**, enter the **Extensional flexibility matrix**, $\mathbf{D}_a$; **Bending–extensional flexibility matrix**, $\mathbf{D}_b$; **Bending flexibility matrix**, $\mathbf{D}_d$; and **Shear flexibility matrix**, $\mathbf{D}_{as}$.

If required, enter also the **Translational inertia**, I_0; **Rotational–translational inertia matrix**, $\mathbf{I}_1$; and **Rotational inertia matrix**, $\mathbf{I}_2$. The translational inertia is the average mass per reference area. As long as the shell is not thick or shear flexible, the two latter contributions to the inertia can usually be ignored. If the mass distribution of the shell is symmetric with respect to the midplane, then $\mathbf{I}_1$ is identically zero.

The default for all section properties is to take the values **From material**. Any one of the matrices can also be **User defined**.

The inertia matrices are needed only for dynamic analysis. They are also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

STRESS EVALUATION PROPERTIES

If you want a certain stress value to be computed, you can enter a linear relation between section forces and stress here.

Enter an **In-plane force factor**, S_N. This is a matrix which transforms the local membrane force (N_{11}, N_{22}, N_{12}) into a local in-plane stress (s_{11}, s_{22}, s_{12}).

Enter a **Moment factor**, S_M. This is a matrix which transforms the local bending moment (M_{11}, M_{22}, M_{12}) into a local in-plane stress (s_{11}, s_{22}, s_{12}).

Enter an **Out-of-plane force factor**, S_Q. This is a matrix which transforms the local shear force (Q_1, Q_2) into a local transverse shear stress (s_{13}, s_{23}).

SHEAR CORRECTION FACTOR

Enter the shear correction factors for transverse shear k_{23} and k_{13}.

When computing the contribution from shear stiffness to the total virtual work, it is necessary to take into account that the shell approximation assumes that shear stresses and strains in the thickness direction are constant, whereas in reality the distribution is more complicated. The shear correction factors are used to compensate for this, so that the total strain energy density is correct. When operating with stiffness and flexibility matrices, this correction can also be built directly into the matrices $\mathbf{D}_{\mathrm{As}}$ and $\mathbf{D}_{\mathrm{as}}$. Is so, set both shear correction factors to 1.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Material Models** > **Section Stiffness**
Plate > **Material Models** > **Section Stiffness**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Material Models** > **Section Stiffness**

Physics tab with **Plate** selected:

Domains > **Material Models** > **Section Stiffness**

Fixed Constraint

The **Fixed Constraint** node adds a condition that makes the geometric entity fixed (fully constrained); that is, the displacements and rotations are zero in all directions.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with weak constraints. In order to be able to interpret the orientations of rotations to which the Lagrange multipliers are defined, you must know to which boundary they are applied. Select the boundary which should be used when setting up the weak constraints. The default is **Use face with lowest number**.

This section is available only for edges and points in the Shell interface and is only visible if **Advanced Physics Options** has been selected.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Face Constraints** > **Fixed Constraint**
Shell > **More Constraints** > **Fixed Constraint** (Edge)
Shell > **Points** > **Fixed Constraint**
Plate > **Face Constraints** > **Fixed Constraint**
Plate > **More Constraints** > **Fixed Constraint** (Boundary)
Plate > **Points** > **Fixed Constraint**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Fixed Constraint**

Edges > **More Constraints** > **Fixed Constraint**
Points > **Shell** > **Fixed Constraint**

Physics tab with **Plate** selected:

Boundaries > **Face Constraints** > **Fixed Constraint**
Edges > **More Constraints** > **Fixed Constraint**
Points > **Plate** > **Fixed Constraint**

Prescribed Displacement/Rotation

The **Prescribed Displacement/Rotation** node adds a point, edge, boundary, or domain condition to a model where the displacements and rotations are prescribed in one or more directions.

With this condition it is possible to prescribe one or more displacement components, as well as one or more rotational components, leaving the shell free to deform or rotate in the remaining directions. It is also possible set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

- Prescribing all displacements and rotations to zero is equivalent to using a **Fixed Constraint**.
- Prescribing all displacements to zero is equivalent to a **Pinned** condition.
- Prescribing all rotations to zero is equivalent to the **No Rotation** condition.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the prescribed displacement/rotation. The coordinate system selection is based on the geometric entity level.

Domains (Plate Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

Boundaries (Shell and Plate Interfaces)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- **Shell Local System**
- Any additional user-defined coordinate system

Edges (Shell Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Local edge system** (the default).
- Any additional user-defined coordinate system.

Points (Shell and Plate Interfaces)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with a **Local edge system** and **Shell Local System**. If displacement or rotations is prescribed for an edge or point which is shared between boundaries, the local system can be ambiguous. Select the boundary which should define the local system. The default is **Use face with lowest number**.

This section is available only for edges and points in the Shell interface.

PRESCRIBED DISPLACEMENT

For the displacement in each direction, select a setting from the list — **Free**, **Prescribed**, or **Limited**. Select:

- **Free** (the default) to leave the displacement component unconstrained.
- **Prescribed** to constrain the displacement component to a given value. Enter a scalar value for the component of the prescribed displacement u_0.
- **Limited** to set a maximum and a minimum limit for the displacement component. Enter values for the maximum displacement $u_{0,\max}$ and the minimum displacement $u_{0,\min}$. By default, they are set to `Inf` and `-Inf`, which corresponds to no active constraint.

If any displacement component is set to **Limited**, an additional section **Limited Displacement** is visible. Select the **Method** used to implement the weak inequality constraint — **Penalty** or **Augmented Lagrangian**. For both methods, enter a **Penalty factor** k_p.

By default, the **Penalty** method is suggested, which in principle enforces the maximum and minimum limits for the displacement by adding nonlinear springs with a stiffness equal to k_p when the limits are exceeded. This method is usually robust, but the accuracy is directly dependent on the chosen penalty factor.

The **Augmented Lagrangian** method adds extra degrees of freedom to improve the accuracy of the constraint. Here, the penalty factor is a numerical parameter, and has less impact on the accuracy of the constraint compared to when using the penalty method. The implementation of the augmented Lagrangian method puts no restrictions on the solver sequence, but for good convergence, proper scaling of the extra degrees of freedom can be important.

The default value for the **Penalty factor** k_p depends on to what type of entity the **Prescribed Displacement/Rotation** node is added. In the expressions below, `<phys>` is the tag of the physics interface, and `<pd>` is the tag of the **Prescribed Displacement/Rotation** node.

- For points, the default expression is `100*<phys>.Eequ*<phys>.<pd>.charLen`
- For edges, the default expression is `100*<phys>.Eequ`
- For boundaries, the default expression is `100*<phys>.Eequ/<phys>.<pd>.charLen`

Here, the terms point, edge, and boundary refer to the actual *physical* entities represented by the chosen *geometric* entities. For instance, in a 2D axisymmetric

component, the circumferential direction is only implicitly modeled, and thus the physical entity level of the shell is interpreted as one level higher than the chosen geometric entity level. The highest entity level in the 2D Plate interface is interpreted as a boundary.

The variable `<phys>.<pd>.charLen` is equal to thickness of the shell or plate.

For details about setting maximum and minimum limits for the displacements, see Limited Displacement

PRESCRIBED ROTATIONS

Select a prescribed rotation from the **By** list — **Free**, **Rotation**, or **Normal vector**. Select:

- **Free** (the default) to leave the rotations unconstrained.
- **Rotation** to activate a prescribed rotation in a certain direction. Enter a value or expression for the prescribed rotation, θ, about the tangential directions **t1** and **t2** of the shell local system. The rotation is always interpreted the with respect to the shell local system, and independent of the Coordinate System Selection.

  For geometric linearity, select one or both of the **Free rotation around t1 direction** and **Free rotation around t2 direction** checkboxes to remove the constraint for the corresponding rotation component. If cleared, the rotations are constrained to either the input value or to the default zero rotation. The status of the checkboxes has no effect when the geometric nonlinearity is activated under the study settings. This is because the constraints for different rotation components are not independent of each other in the case of finite rotations.

In 2D axisymmetry, there is only one input, the **Prescribed rotation** around the out-of-plane direction, Θ.

- **Normal vector** to describe the rotation by prescribing the shell normal vector in the deformed configuration. Enter the components of the **Prescribed normal vector** N_0.

For details, see Initial Values and Prescribed Values.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

When **Individual dependent variables** is selected in the **Apply reaction terms on** list, the constraint forces are applied directly on the degrees of freedom, which are the displacements along the global coordinate axes. If you use this setting together with a local coordinate system, the results will be inconsistent since the constraint forces will not match the constraint orientation.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.
- You can assign the value of the prescribed displacement to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Face Constraints** > **Prescribed Displacement/Rotation**
Shell > **Prescribed Displacement/Rotation** (Edge)
Shell > **Points** > **Prescribed Displacement/Rotation**
Plate > **Face Constraints** > **Prescribed Displacement/Rotation**
Plate > **Prescribed Displacement/Rotation** (Boundary)
Plate > **Points** > **Prescribed Displacement/Rotation**

Ribbon
Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Prescribed Displacement/Rotation**
Edges > **Shell** > **Prescribed Displacement/Rotation**
Points > **Shell** > **Prescribed Displacement/Rotation**

Physics tab with **Plate** selected:

Domains > **Face Constraints** > **Prescribed Displacement/Rotation**
Boundaries > **Plate** > **Prescribed Displacement**
Points > **Plate** > **Prescribed Displacement**

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Fixed Constraint**, **Prescribed Displacement**, **Pinned**, or **No Rotation**) to prescribe a deformation of the constraint caused by changes in temperature of the surroundings. This makes it possible to reduce stresses caused by the boundary conditions.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of the surroundings idealized by the constraints.

Select **Inherit from boundary** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the domain itself.

When **Inherit from boundary** is not selected, enter:

- A value or expression for the **Volume reference temperature** T_{ref} that is the temperature at which there are no thermal displacements at the constraints.
- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any temperature variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The

default values **From material** are used. This requires that a material has been assigned to the boundaries, edges, or points where the constraint is active. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

Enter the coordinates of the **Reference point**, which is the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Thermal Expansion Properties.
- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Fixed Constraint** > **Thermal Expansion**
Shell > **Prescribed Displacement** > **Thermal Expansion**
Shell > **Pinned** > **Thermal Expansion**
Shell > **No Rotation** > **Thermal Expansion**
Plate > **Fixed Constraint** > **Thermal Expansion**
Plate > **Prescribed Displacement** > **Thermal Expansion**
Plate > **Pinned** > **Thermal Expansion**
Plate > **No Rotation** > **Thermal Expansion**

Ribbon

Physics tab with **Fixed Constraint**, **Prescribed Displacement**, **Pinned**, or **No Rotation** node selected in the model tree:

Attributes > **Thermal Expansion**

Prescribed Velocity

The **Prescribed Velocity** node adds an edge, boundary, or domain condition where the translational or rotational velocity is prescribed in one or more directions. The prescribed velocity condition is applicable for Time Dependent and Frequency

Domain studies. With this condition it is possible to prescribe a velocity in one direction, leaving the shell free in the other directions.

The **Prescribed Velocity** node is a constraint, and overrides any other constraint on the same selection.

For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the prescribed velocity. The components are entered in the selected coordinate system.

The coordinate system selection is based on the geometric entity level.

Coordinate systems with directions that change with time should not be used. If you choose another, local coordinate system, the velocity components change accordingly.

Domains (Plate Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

Boundaries (Shell and Plate Interfaces)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- **Shell Local System**
- Any additional user-defined coordinate system

Edges (Shell Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Local edge system** (the default)

- **Shell Local System**
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with a **Local edge system** and **Shell Local System**. If the velocity is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface.

PRESCRIBED VELOCITY

To define a prescribed velocity for each spatial direction (x, y, and z), select one or more of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** checkboxes. Then enter a value or expression for the prescribed velocity components v_x, v_y, and v_z.

PRESCRIBED ANGULAR VELOCITY

To define a prescribed angular velocity for each spatial direction (x, y, and z), select one or all of the **Prescribed around x direction**, **Prescribed around y direction**, and **Prescribed around z direction** checkboxes and enter a value or expression for in each $\partial\theta_x/\partial t$, $\partial\theta_y/\partial t$, or $\partial\theta_z/\partial t$ field.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed velocity in a frequency domain analysis of perturbation type.
- **Prescribed Velocity** cannot be used as a weak constraint.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Face Constraints** > **Prescribed Velocity**

Shell > **More Constraints** > **Prescribed Velocity** (Edge)
Plate > **Face Constraints** > **Prescribed Velocity**
Plate > **More Constraints** > **Prescribed Velocity** (Boundary)

Ribbon
Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Prescribed Velocity**
Edges > **More Constraints** > **Prescribed Velocity**

Physics tab with **Plate** selected:

Domains > **Face Constraints** > **Prescribed Velocity**
Boundaries > **More Constraints** > **Prescribed Velocity**

Prescribed Acceleration

The **Prescribed Acceleration** node adds an edge, boundary, or domain condition where the translational or rotational acceleration is prescribed in one or more directions. The prescribed acceleration condition is applicable for Time Dependent and Frequency Domain studies. With this condition it is possible to prescribe an acceleration in one direction, leaving the shell free in the other directions.

The **Prescribed Acceleration** node is a constraint, and overrides any other constraint on the same selection.

For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the prescribed acceleration. The coordinate system selection is based on the geometric entity level.

Coordinate systems with directions that change with time should not be used. If you choose another, local coordinate system, the acceleration components change accordingly.

Domains (Plate Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

Boundaries (Shell and Plate Interfaces)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- **Shell Local System**
- Any additional user-defined coordinate system

Edges (Shell Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Local edge system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with a **Local edge system** and **Shell Local System**. If the acceleration is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface.

PRESCRIBED ACCELERATION

To define a prescribed acceleration for each spatial direction (x, y, and z), select one or more of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** checkboxes. Then enter a value or expression for the prescribed acceleration components a_x, a_y, and a_z.

PRESCRIBED ANGULAR ACCELERATION

To define a prescribed angular acceleration for each spatial direction (x, y, and z), select one or all of the **Prescribed around x**, **y**, and **z direction** checkboxes and enter a value or expression for in each $\partial^2\theta_x/\partial t^2$, $\partial^2\theta_y/\partial t^2$, or $\partial^2\theta_z/\partial t^2$ field.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed acceleration in a frequency domain analysis of perturbation type.
- **Prescribed Acceleration** cannot be used as a weak constraint.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Face Constraints** > **Prescribed Acceleration**
Shell > **More Constraints** > **Prescribed Acceleration** (Edge)
Plate > **Face Constraints** > **Prescribed Acceleration**
Plate > **More Constraints** > **Prescribed Acceleration** (Boundary)

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Prescribed Acceleration**
Edges > **More Constraints** > **Prescribed Acceleration**

Physics tab with **Plate** selected:

Domains > **Face Constraints** > **Prescribed Acceleration**
Boundaries > **More Constraints** > **Prescribed Acceleration**

Pinned

The **Pinned** node adds an edge, boundary, domain, or point condition that fixes the translations in all directions; that is, all displacements are zero. The rotations are not constrained.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Face Constraints** > **Pinned**
Shell > **More Constraints** > **Pinned** (Edge)
Shell > **Points** > **Pinned**
Plate > **Face Constraints** > **Pinned**
Plate > **More Constraints** > **Pinned** (Boundary)
Plate > **Points** > **Pinned**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Pinned**
Edges > **More Constraints** > **Pinned**
Points > **Shell** > **Pinned**

Physics tab with **Plate** selected:

Boundaries > **Face Constraints** > **Pinned**
Edges > **More Constraints** > **Pinned**
Points > **Plate** > **Pinned**

No Rotation

The **No Rotation** node adds an edge, boundary, domain, or point condition that fixes the rotations around all axes. The translations are not constrained.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with weak constraints. In order to be able to interpret the orientations to which the Lagrange multipliers are defined, you must know to which boundary they are applied. Select the boundary which should be used when setting up the weak constraints. The default is **Use face with lowest number**.

This section is available only for edges and points in the Shell interface and is only visible if **Advanced Physics Options** has been selected.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Face Constraints** > **No Rotation**
Shell > **More Constraints** > **No Rotation** (Edge)
Shell > **Points** > **No Rotation**
Plate > **Face Constraints** > **No Rotation**
Plate > **More Constraints** > **No Rotation** (Boundary)
Plate > **Points** > **No Rotation**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **No Rotation**
Edges > **More Constraints** > **No Rotation**
Points > **Shell** > **No Rotation**

Physics tab with **Plate** selected:

Boundaries > **Face Constraints** > **No Rotation**
Edges > **More Constraints** > **No Rotation**
Points > **Plate** > **No Rotation**

Simply Supported

The **Simply Supported** node adds an edge condition that constrains the displacement in the direction perpendicular to the shell. The in-plane rotation perpendicular to the edge is also constrained. Optionally you can constrain the in-plane displacements.

IN-PLANE DISPLACEMENT CONSTRAINTS

Select **Along edge** to constrain the in-plane translation along the edge.

Select **Perpendicular to edge** to constrain the in-plane displacement perpendicular to the edge.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **More Constraints** > **Simply Supported**

Plate > **More Constraints** > **Simply Supported**

Ribbon

Physics tab with **Shell** selected:

Edges > **More Constraints** > **Simply Supported**

Physics tab with **Plate** selected:

Boundaries > **More Constraints** > **Simply Supported**

Symmetry

The **Symmetry** node adds an edge or boundary condition that defines a symmetry edge or boundary.

For 2D axisymmetric components, use the Symmetry Plane node instead.

COORDINATE SYSTEM SELECTION (EDGES ONLY)

This section is available only for edges in the Shell interface.

Select the coordinate system to use for specifying a symmetry edge. From the **Coordinate system** list select from:

- **Local edge system** (the default).
- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This section is available only for edges in the Shell interface.

This setting is used in conjunction with a **Local edge system** and **Shell Local System**. If symmetry is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

SYMMETRY

This section is available only for edges in the Shell interface.

If another coordinate system than the **Local edge system** is used, select a **Symmetry plane normal** — **First axis**, **Second axis**, or **Third axis**. This indicates which axis in the selected coordinate system that acts as normal to the symmetry plane.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_n. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.shell.sym1.un`.
- In the Plate interface, this section is only present when **Use 3D formulation** is selected in the interface settings.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry and Antisymmetry Boundary Conditions

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Face Constraints** > **Symmetry**

Shell > **More Constraints** > **Symmetry** (Edge)
Plate > **More Constraints** > **Symmetry** (Boundary)

Ribbon
Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Symmetry**
Edges > **More Constraints** > **Symmetry**

Physics tab with **Plate** selected:

Boundaries > **More Constraints** > **Symmetry**

Symmetry Plane

This node is only available in the 2D axisymmetric Shell interface.

In an axially symmetric problem, the only possible symmetry plane has the Z-axis as normal. Use the **Symmetry Plane** node to impose such a symmetry condition.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_{n}. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.shell.symp1.un`.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **More Constraints** > **Symmetry Plane**

Ribbon

Physics tab with **Shell** selected:

Points > **More Constraints** > **Symmetry Plane**

Antisymmetry

The **Antisymmetry** node adds an edge or boundary condition that defines an antisymmetry edge or boundary.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with a **Local edge system**. If antisymmetry is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface.

COORDINATE SYSTEM SELECTION (EDGES ONLY)

Select the coordinate system to use for specifying a symmetry edge. From the **Coordinate system** list select from:

- **Local edge system** (the default).
- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

ANTISYMMETRY (EDGES ONLY)

If another coordinate system than the **Local edge system** is used, select a **Symmetry plane normal** — **First axis**, **Second axis**, or **Third axis**. This indicates which axis in the selected coordinate system that acts as normal to the antisymmetry plane.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry and Antisymmetry Boundary Conditions

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Face Constraints** > **Antisymmetry**
Shell > **More Constraints** > **Antisymmetry** (Edge)
Plate > **More Constraints** > **Antisymmetry** (Boundary)

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face Constraints** > **Antisymmetry**
Edges > **More Constraints** > **Antisymmetry**

Physics tab with **Plate** selected:

Boundaries > **More Constraints** > **Antisymmetry**

Body Load

Add a **Body Load** to boundaries (for the Plate interface add it to domains). The loads are defined in the given coordinate system.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select the **Load type** — **Force per reference volume**, **Force per deformed volume**, or **Total force**. Enter values or expressions for $\mathbf{f}_V$, $\mathbf{f}_v$, or $\mathbf{F}_{tot}$.

The list selection normally only contains **User defined**. When combining the Shell interface with another physics interface, it is also possible to select a predefined load from this list.

MOMENT

Select the **Moment type** — **Moment per reference volume**, **Moment per deformed volume**, or **Total moment**. Enter values or expressions for $\mathbf{m}_V$, $\mathbf{m}_v$, or $\mathbf{M}_{tot}$.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Face and Volume Loads** > **Body Load**

Plate > **Face and Volume Loads** > **Body Load**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Face and Volume Loads** > **Body Load**

Physics tab with **Plate** selected:

Domains > **Face and Volume Loads** > **Body Load**

Face Load

Add a **Face Load** to boundaries (for the Plate interface add it to domains), to use it as a pressure or tangential force acting on a surface. The loads are defined in the given coordinate system.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems in the *COMSOL Multiphysics Reference Manual*.

THROUGH-THICKNESS LOCATION

Select a surface — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is twofold:

- For a curved boundary, the difference in area between the midsurface and the selected surface is taken into account.
- For a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

FORCE

Select a **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, **Pressure**, or **Resultant**. The last option is only available in the Shell interface, and only in 3D components.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.
- For **Total force**, COMSOL Multiphysics divides the total force by the area of the boundaries where the load is active. Then the force is applied in the same way as for a **Force per reference area**. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Pressure**, a scalar input is given, and the orientation of the load is given by the normal to the boundary. The pressure is positive when directed in the opposite direction of the normal to the shell. In a geometrically nonlinear analysis, the current surface normal and area are used.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected boundaries. For **Point**, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

For the load types **Force per reference length** and **Total force**, the load list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference area	$\mathbf{f}_A$	N/m^2	boundaries domains	3D (x, y, z) 2D axisymmetric (r, z) 2D (x, y, z)
Force per deformed area	$\mathbf{f}_a$	N/m^2	boundaries domains	3D (x, y, z) 2D axisymmetric (r, z) 2D (x, y, z)
Total force	$\mathbf{F}_{tot}$	N	boundaries domains	3D(x, y, z) 2D axisymmetric (r, z) 2D(x, y, z)

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Pressure	p	Pa	boundaries	3D
			domains	2D axisymmetric
				2D
Resultant	**F**, **M**	N, N m	boundaries	3D (x, y, z)

MOMENT

Select the **Moment type** — **Moment per reference area**, **Moment per deformed area**, or **Total moment**. Enter values or expressions for $\mathbf{m}_A$, $\mathbf{m}_a$, or $\mathbf{M}_{tot}$.

This section is not present when **Resultant** is selected as the **Load type**.

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and `<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the

loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Face and Volume Loads** > **Face Load**
Plate > **Face and Volume Loads** > **Face Load**
Membrane > **Face and Volume Loads** > **Face Load**

Ribbon

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Face and Volume Loads** > **Face Load**

Physics tab with **Plate** selected:

Domains > **Face and Volume Loads** > **Face Load**

Edge Load

Add an **Edge Load** as a force or moment distributed along an edge (for the Plate interface add it to boundaries). The load is defined in the given coordinate system.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the load. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Shell Local System**
- **Local edge system**

For details about the definition of local edge systems, see Local Edge System.

- Any additional user-defined coordinate system.

FACE DEFINING THE ORIENTATIONS

This setting is used in conjunction with **Local edge system** and **Shell Local System**. When the load is applied to an edge which is shared between boundaries, the coordinate system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface and is only visible if the selected coordinate system is **Local edge system** or **Shell Local System**.

THROUGH-THICKNESS LOCATION

Select a surface — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is twofold:

- For a curved boundary, the difference in area between the midsurface and the selected surface is taken into account.
- For a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

If a selected edge is connected to boundaries having different thicknesses, then the result of specifying **Top Surface** or **Bottom Surface** is not well defined.

FORCE

Select a **Load type** — **Force per reference area**, **Force per deformed area**, **Force per reference length**, **Force per deformed length**, **Total force**, or **Resultant**.

- For **Force per reference length** and **Force per deformed length**, the traction components are given explicitly.
- For **Force per reference area** and **Force per deformed area**, the given traction components are multiplied by the thickness of the shell in order to give the edge load.
- For **Total force**, COMSOL Multiphysics divides the total force by the length of the edges where the load is active. Then the force is applied in the same way as for a **Force per reference length**. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected boundaries. For **Point**, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

The load list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Force per reference length	$\mathbf{f}_L$	N/m	edges	3D
			boundaries	2D
Force per deformed length	$\mathbf{f}_l$	N/m	edges	3D
			boundaries	2D
Force per reference area	$\mathbf{f}_A$	N/m^2	edges	3D
			boundaries	2D
Force per deformed area	$\mathbf{f}_a$	N/m^2	edges	3D
			boundaries	2D

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Total force	$\mathbf{F}_{tot}$	N	edges	3D
			boundaries	2D
Resultant	**F**, **M**	N, N m	edges	3D

MOMENT

Select a **Moment type** to define the moment load — **Moment per reference area**, **Moment per deformed area**, **Moment per reference length**, **Moment per deformed length**, or **Total Moment**. Enter values or expressions for $\mathbf{m}_A$, $\mathbf{m}_a$, $\mathbf{m}_L$, $\mathbf{m}_l$, or $\mathbf{M}_{tot}$.

This section is available only in the Shell and Plate interfaces. It is not present when **Resultant** is selected as the **Load type**.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Moment per reference area	$\mathbf{m}_A$	N	edges	3D
			boundaries	2D
Moment per deformed area	$\mathbf{m}_a$	N	edges	3D
			boundaries	2D
Moment per reference length	$\mathbf{m}_L$	N	edges	3D
			boundaries	2D
Moment per deformed length	$\mathbf{m}_l$	N·m/m^2	edges	3D
			boundaries	2D
Total moment	$\mathbf{M}_{tot}$	N·m	edges	3D
			boundaries	2D

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and

<phys_tag>.<load_tag>.x3, for example solid.bndl1.x2. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Edge Load**
Plate > **Edge Load**
Membrane > **Edge Load**

Ribbon

Physics tab with **Shell** or **Membrane** selected:

Boundaries > **Shell** > **Edge Load**
Boundaries > **Membrane** > **Edge Load**

Physics tab with **Plate** selected:

Domains > **Plate** > **Edge Load**

Point Load

Add a **Point Load** to points for concentrated forces or moments at points. The loads are defined in the given coordinate system. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Shell Local System**
- Any additional user-defined coordinate system.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with **Shell Local System**. When the load is applied to an edge which is shared between boundaries, the coordinate system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface and is only visible if the selected coordinate system is **Shell Local System**.

THROUGH-THICKNESS LOCATION

Select a surface — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is that for a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

If a selected point is connected to boundaries having different thicknesses, then the result of specifying **Top Surface** or **Bottom Surface** is not well defined.

FORCE

Select a **Load type** — **Force per point**, **Total force**, or **Resultant**.Then enter values or expressions based on the selection.

- For **Force per point** enter values or expressions for the components of the point load $\mathbf{F}_p$.
- For **Total force**, COMSOL Multiphysics divides the total force by the number of points where the load is active. Then the force is applied in the same way as for **Force per point**.
- For **Resultant**, enter the **Force** and **Moment** with respect to a point. Select the **Application point defined using** — **Centroid**, **Point**, or **Coordinates**. For **Centroid**, the coordinates of the application point is the centroid of the selected points. For **Point**, select a geometrical point in the section **Application Point**. For **Coordinates**, enter the global coordinates of the **Application point**, $\mathbf{x}_a$.

For the load types **Force per point** and **Total force**, the load list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

MOMENT

Select a **Moment type** to define the moment load — **Moment per point** or **Total moment**. Enter values or expressions for $\mathbf{M}_P$ or $\mathbf{M}_{tot}$. The given moment is applied to each of the selected points.

This section is not present when **Resultant** is selected as the **Load type**.

This section is only shown when **Resultant** has been selected as **Load type**. The distribution of the forces over the selected points is controlled by the settings here.

Select a **Traction distribution** — **Beam** or **User defined**.

When **Beam** is selected, the force distribution approximately matches the stress distribution over a beam cross section.

For **User Defined**, you can write expressions for the force distribution. For 3D components enter expressions for the six dimensionless vector-valued distribution functions, $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$, $\mathbf{q}_5$, and $\mathbf{q}_6$. For 2D components enter expressions for $\mathbf{q}_1$, $\mathbf{q}_2$,

and $\mathbf{q}_4$. Usually, the local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the functions. The built-in variables for the local coordinates are named `<physics_tag>.<load_tag>.x2` and `<phys_tag>.<load_tag>.x3`, for example `solid.bndl1.x2`. The default value is the distribution that is used when **Beam** is selected as **Traction distribution**.

The local coordinates, x_2 and x_3, are defined in the following way:

- The origin is placed at the centroid of the selected points.
- The axis directions coincide with the principal axes of a virtual collection of points with unit mass.
- x_2 is the axis around which the virtual principal area moment of inertia is larger.

When **User Defined** is selected, you can also select a **Weight function** — **None**, **Circular**, or **User Defined**. This function acts as a multiplier to the force distribution functions. When **None** is selected, the weight function $w = 1$. For **Circular**, enter a radius r_w of a circle, outside of which no load is applied ($w = 0$). Inside the circle, $w = 1$. For **User Defined**, enter a weighting expression. Usually, the built-in local coordinates of the loaded region would be used for this purpose, but there is no limitation on the form of the function. A time- or parameter-dependent function can, for example, be used to model a moving load.

The traction distribution functions have the property that the forces are distributed as

$$\mathbf{f}(\mathbf{x}) = w(\mathbf{x}) \sum_{i=1}^{6} c_i \mathbf{q}_i(\mathbf{x})$$

where the coefficients c_i are chosen so that the given force and moment resultants are obtained.

SYMMETRY

This section is only shown when **Resultant** has been selected as **Load type**. If the resultant force is applied in a symmetry or antisymmetry plane, this fact needs to be taken into account when creating the corresponding force distribution.

Select a symmetry type — **None**, **Symmetry**, or **Antisymmetry**. When one of the types of symmetry is chosen, specify the symmetry plane by entering a **Normal Vector**, $\mathbf{n}_{\text{sym}}$, and a **Point**, $\mathbf{x}_{\text{sym}}$, located in the plane.

When specifying a resultant load in a symmetry plane, you still provide the full value of force and moment, as if there were no symmetry in the model. Any components of the given load that do not fulfill the selected type of symmetry will be discarded.

When using the **Resultant** option, the following should be observed

- In 3D, the selection must contain at least three points, and all points cannot be collinear.
- in 2D, the selection must contain at lease two points.
- While it is formally possible to select points that are not located in a plane, the force distribution may be unpredictable in such a case.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Points** > **Point Load**

Plate > **Points** > **Point Load**

Ribbon

Physics tab with **Shell** or **Plate** selected:

Points > **Point Load**

Point Load, Free

Add a **Point Load, Free** node to describe concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Imported meshes, where there may not be suitable points for load application
- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free point load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable `<phys>.<load_tag>.num_ignored`, for example `shell.plf1.num_ignored`.
- The local stress field in the element where a point load acts will not be reliable.

LOCATION AND FORCE

Select a **Through-thickness location** — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is that for a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

The coordinates that you enter for the load location are not affected by this setting; they should always refer to the reference surface, that is the coordinates of the mesh.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

Select a **Frame** — **Material** or **Spatial** that determines how the given coordinates are interpreted in case of a geometrically nonlinear analysis. When **Material** is selected, the load location on the body is interpreted with respect to the initial configuration. When **Spatial** is selected, then the load location is given in space fixed coordinates, which usually means that the material point on the body where the load is applied will change with deformation even if the coordinate values are constant.

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**X**, **Y**, **Z**). If spatial frame input is used, then the spatial coordinates (**x**, **y**, **z**) are used. Then, enter the force and moment values, **Fxl**, **Fyl**, **Fzl**, **Mxl, Myl**, and **Mzl**. The force and moment vectors are interpreted in the coordinate system selected in the **Coordinate System Selection** section.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

Pratt Truss Bridge: Application Library path
Structural_Mechanics_Module/Beams_and_Shells/pratt_truss_bridge

Context Menus
Shell > **Point Load, Free**
Plate > **Point Load, Free**

Ribbon
Physics tab with **Shell** selected:

Global > **Shell** > **Point Load, Free**

Physics tab with **Plate** selected:

Global > **Plate** > **Point Load, Free**

Ring Load

A load applied to a point, not located at the axis of revolution, in an axisymmetric model actually represents a ring load. Add a **Ring Load** to points located at $R > 0$ to model such loads.

THROUGH-THICKNESS LOCATION

Select a surface — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is that for a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

If a selected point is connected to boundaries having different thicknesses, then the result of specifying **Top Surface** or **Bottom Surface** is not well defined.

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for $\mathbf{f}_A$, $\mathbf{f}_a$, $\mathbf{f}_L$, $\mathbf{f}_l$, or $\mathbf{F}_{tot}$.

MOMENT

Select the **Moment type** — **Moment per reference area**, **Moment per deformed area**, **Moment per reference length**, **Moment per deformed length**, or **Total moment**. Enter values or expressions for $\mathbf{m}_A$, $\mathbf{m}_a$, $\mathbf{m}_L$, $\mathbf{m}_l$, or $\mathbf{M}_{tot}$.

- The **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.
- The **Total force** and **Total moment** options should not be directly interpreted as resultants, since the orientation is changing. Rather, it should be interpreted as the per reference length values, multiplied by $2\pi r$, where r is the radius of the point.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Ring Load**

Ribbon

Physics tab with **Shell** selected:

Points > **Shell** > **Ring Load**

Ring Load, Free

A load applied to a point, not located at the axis of revolution, in an axisymmetric model actually represents a ring load. Add a **Ring Load, Free** node to describe concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Imported meshes, where there may not be suitable points for load application

- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable `<phys>.<load_tag>.num_ignored`, for example `shell.rlf1.num_ignored`.
- The local stress field in the element where a point load acts will not be reliable.

LOCATION AND FORCE

Select a **Through-thickness location** — **Top Surface**, **Midsurface**, or **Bottom Surface**. The default is that the load is applied at the midsurface. The effect of using another surface than the midsurface is that for a tangential load, the distance from the midsurface is used to compute an additional equivalent moment load.

To place the load at another distance from the midsurface, select the **Offset** checkbox, and enter a value for the offset, z_{offset}.

The coordinates that you enter for the load location are not affected by this setting; they should always refer to the reference surface, that is the coordinates of the mesh.

If the material model is **Section Stiffness**, there may physically not be a well-defined top and bottom surface. If the load is applied at such a surface, the thickness value used to compute the load is taken from the settings in the **Thickness and Offset** node.

Select a **Frame** — **Material** or **Spatial** that determines how the given coordinates are interpreted in case of a geometrically nonlinear analysis. When **Material** is selected, the load location on the body is interpreted with respect to the initial configuration. When **Spatial** is selected, then the load location is given in space fixed coordinates, which

usually means that the material point on the body where the load is applied will change with deformation even if the coordinate values are constant.

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**R**, **Z**). If spatial frame input is used, then the spatial coordinates (**r**, **z**) are used. Then, enter the force and moment values, **FxI**, **FzI**, and **MyI**. The force and moment components are interpreted in the coordinate system selected in the **Coordinate System Selection** section.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

Location in User Interface

Context Menus

Shell > **Ring Load, Free**

Ribbon

Physics tab with **Shell** selected:

Global > **Shell** > **Ring Load, Free**

Spring Foundation

Use the **Spring Foundation** node to apply elastic and damping boundary conditions for domains, boundaries, edges, and points.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the spring properties. The coordinate system selection is based on the geometric entity level.

Domains (Plate Interface)

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

Boundaries (Shell and Plate Interfaces)

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- **Shell Local System**
- Any additional user-defined coordinate system

Edges (Shell Interface)

From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Local edge system** (the default).
- **Shell Local System**
- Any additional user-defined coordinate system.

Points (Shell and Plate Interfaces)

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

For details about the definition of local edge systems, see Local Edge System.

SPRING

Select the **Spring type** and its associated spring constant or force using Table 6-22 as a guide. The default option is the spring type for the type of geometric entity and space dimension, and there are different combinations available based on this.

When the option is of the type 'force as function of extension', then the built-in variables describing the spring extension must be used in the expression as described in Springs and Dampers. The spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

TABLE 6-22: SPRING TYPES FOR THE SPRING FOUNDATION FEATURE.

SPRING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Spring constant per reference volume	$\mathbf{k}_V$	$N/(m \cdot m^3)$	domains (Plate), boundaries (Shell)
Total spring constant	$\mathbf{k}_{tot}$	N/m	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Spring constant per reference area	$\mathbf{k}_A$	$N/(m \cdot m)^2$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Spring constant per reference length	$\mathbf{k}_L$	$N/(m \cdot m)$	boundaries (Plate), edges (Shell)
Spring constant	$\mathbf{k}_P$	N/m	points (Shell, Plate)
Force per volume as function of extension	$\mathbf{F}_V$	N/m^3	domains (Plate), boundaries (Shell)
Total force as function of extension	$\mathbf{F}_{tot}$	N	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Force per length as function of extension	$\mathbf{F}_L$	N/m	boundaries (Plate), edges (Shell)
Force as function of extension	$\mathbf{F}_P$	N	points (Shell, Plate)

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s which is used to multiply all values of the spring matrix or spring force vector.

- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full**, then enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS DAMPING

Select the **Damping type** using Table 6-23 as a guide. The default option is the default damping type for the type of geometric entity and space dimension, and there are different combinations available based on this. The damping matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 6-23: DAMPING TYPES FOR THE SPRING FOUNDATION FEATURE.

DAMPING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Damping constant per reference volume	$\mathbf{d}_V$	N·s/(m·m^3)	domains (Plate), boundaries (Shell)
Damping constant per reference area	$\mathbf{d}_A$	N·s/(m·m^2)	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Total damping constant	$\mathbf{d}_{tot}$	N·s/m	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Damping constant per reference length	$\mathbf{d}_L$	N·s/(m·m)	boundaries (Plate), edges (Shell)

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

- Springs and Dampers in the Structural Mechanics Modeling chapter.
- Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Mass, Spring, and Damper** > **Spring Foundation** (Boundary, Edge)
Shell > **Points** > **Spring Foundation**
Plate > **Mass, Spring, and Damper** > **Spring Foundation** (Domain, Boundary)
Plate > **Points** > **Spring Foundation**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Mass, Spring, and Damper** > **Spring Foundation**
Points > **Shell** > **Spring Foundation**
Points > **Membrane** > **Spring Foundation**

Physics tab with **Plate** selected:

Domains > **Mass, Spring, and Damper** > **Spring Foundation**
Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Points > **Plate** > **Spring Foundation**

Predeformation

Use the **Predeformation** subnode to specify that the elastic forces in Spring Foundation are nonzero at zero displacement. Thus, you can model cases where the unstressed state of the spring is in another configuration than the one described by the geometry.

The value of the predeformation can vary during the simulation, so it should not be interpreted as an initial value.

SPRING PREDEFORMATION

Based on space dimension, enter the values for the **Spring Predeformation** $\mathbf{u}_0$.

You can assign the load caused by the predeformation to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Spring Foundation** > **Predeformation**

Plate > **Spring Foundation** > **Predeformation**

Ribbon
Physics tab with **Spring Foundation** node selected in the model tree:

Attributes > **Predeformation**

Added Mass

The **Added Mass** node is available on boundaries and edges and can be used to supply inertia that is not part of the material itself. Such inertia does not need to be isotropic, in the sense that the inertial effects are not the same in all directions.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the mass distribution in case of a nonisotropic mass contribution. The coordinate system selection is based on the geometric entity level.

Domains (Plate Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Shell Local System**
- Any additional user-defined coordinate system

Boundaries (Shell and Plate Interfaces)
From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- **Shell Local System**
- Any additional user-defined coordinate system

Edges (Shell Interface)
From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Local edge system** (the default)

- **Shell Local System**
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This setting is used in conjunction with a **Local edge system** and **Shell Local System**. If the mass is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface.

All other settings are described in the documentation for Added Mass in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Mass, Spring, and Damper** > **Added Mass** (Boundary, Edge)
Plate > **Mass, Spring, and Damper** > **Added Mass** (Domain, Boundary)

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Mass, Spring, and Damper** > **Added Mass**

Physics tab with **Plate** selected:

Domains > **Mass, Spring, and Damper** > **Added Mass**
Boundaries > **Mass, Spring, and Damper** > **Added Mass**

Point Mass

Use the **Point Mass** node to model a discrete mass or mass moment of inertia that is concentrated at a point.

COORDINATE SYSTEM SELECTION

With the **Coordinate system** list, control the coordinate system around the axis of which the principal mass moments of inertia are defined.

POINT MASS

Enter a **Point mass** m.

Enter a single value for an isotropic **Mass moment of inertia J**, or select **Diagonal** or **Symmetric** to enter a full moment of inertia tensor.

DAMPING

Enter a **Mass damping parameter** α_{dM}. This is the mass proportional term of a Rayleigh damping.

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Points** > **Point Mass**
Plate > **Points** > **Point Mass**

Ribbon

Physics tab with **Shell** or **Plate** selected:

Points > **Shell** > **Point Mass**
Points > **Plate** > **Point Mass**

Periodic Condition

Use a **Periodic Condition** to prescribe that the displacements and rotations in two sets of entities with the same geometrical shape are related as in a periodic structure. In 3D edges can be connected with a periodic condition, and in 2D axisymmetry points. In the Plate interface, the condition can be set up for edges (boundaries).

Several different periodicity types can be prescribed using this boundary condition.

- The **Continuity**, **Antiperiodicity**, and **User defined** periodic conditions directly prescribe relations both between the displacements and between the rotations. They can be used for any study type.
- The **Floquet periodicity** can be used for frequency domain problems with a spatial periodicity of the geometry and solution. The modeled structure is typically a unit cell of a repetitive structure. It can also be used for linear buckling analysis of such structures.
- The **Cyclic symmetry** is a special case of a Floquet condition, intended for structures, which consist of a number of sectors that are identical when rotated around a common axis, like in a fan. It can be used in frequency domain studies, as well as for linear buckling analysis.

The two sets of edges or points between which there is a periodicity condition are called the *source* and *destination*. It is not required to have the same mesh on the

source and destination, but the local accuracy of the solution near the edges will be better if you use the same mesh.

For periodic conditions on shells, the periodicity condition acts on edges or points, as opposed to solids and plates where it acts on boundaries. This means that the orientation cannot be determined automatically. You must provide coordinate systems using the **Orientation of Source** and **Orientation of Destination** sections. The default coordinate system is the Global coordinate system, which works well if the edges are parallel. In other cases, you need to add a **Destination Selection** subnode, in order to supply the coordinate system for the destination.

In cases of rotational symmetry, you can assign the same cylindrical coordinate system to both source and destination. Note that cylindrical coordinate systems in general cannot be used in situations where they are accessed in points located on the axis of revolution, since axis orientations are not uniquely defined there. Thus, if the edges intersect at the axis of revolution, you must use two different coordinate systems with fixed axis directions, rotated with respect to each other.

When there is a common point on the axis of revolution, you should explicitly constrain it to remain on the axis and to have no rotations using **Prescribed Displacement/Rotation**.

EDGE SELECTION/POINT SELECTION/BOUNDARY SELECTION

Select both the source and destination edges or points.

The software automatically identifies the edges as either source edges or destination edges. This works fine for cases like opposing parallel edges. In more general cases, right-click the Period Condition node and choose Manual Destination Selection to add a **Destination Selection** section where you can specify the edges which constitute the destination. By default, this node contains the selection that COMSOL Multiphysics has identified.

In cases where the periodic edge is split into several edges within the geometry, it might be necessary to apply separate periodic conditions to each pair of geometric edges for the matching to work properly.

In the 2D axisymmetry, **Edge Selection** is replaced by **Point Selection**. In the Plate interface, **Edge Selection** is replaced by **Boundary Selection**.

DESTINATION SELECTION

This section is available for specifying the destination edges, points, or boundaries, if needed, when the **Manual Destination Selection** option is selected in the context menu for the **Periodic Condition** node. You can only select destination edges or boundaries from the union of all source and destination edge or boundaries.

PERIODICITY SETTINGS

Select **Type of periodicity** — **Continuity**, **Antiperiodicity**, **Floquet periodicity**, **Cyclic symmetry**, or **User defined**.

- For **Continuity** the displacements and rotations on the destination are set equal to their counterparts on the source; $\mathbf{u}(\mathbf{x}_d) = \mathbf{u}(\mathbf{x}_s)$ and $\mathbf{a}(\mathbf{x}_d) = \mathbf{a}(\mathbf{x}_s)$. If the source and destination objects are rotated with respect to each other, a transformation is performed using the selected coordinate systems, so that corresponding components of the degrees of freedom are connected.
- For **Antiperiodicity** the displacements and rotations on the destination are set equal to their counterparts on the source but with the sign reversed; $\mathbf{u}(\mathbf{x}_d) = -\mathbf{u}(\mathbf{x}_s)$ and $\mathbf{a}(\mathbf{x}_d) = -\mathbf{a}(\mathbf{x}_s)$. If the source and destination objects are rotated with respect to each other, a transformation is performed using the selected coordinate systems, so that corresponding components of the degrees of freedom are connected.
- For **Floquet periodicity** enter a **k-vector for Floquet periodicity** $\mathbf{k}_F$. This is the wave number vector for the excitation.
- For **Cyclic symmetry** the settings differ slightly between the Plate and Shell interfaces.

In the Plate interface, choose how to define the sector angle that the geometry represents using the **Sector angle** list. If **Automatic** is selected, the program attempts to find out how many full repetitions of the geometry there will be on a full revolution. If **User defined** is selected, enter a value for the sector angle θ_S.

In the Shell interface, enter the **Axis direction** vector, $\mathbf{t}_c$, if any point on the edge selection is located on the cyclic symmetry axis. The orientation of the cyclic symmetry axis is then needed for eliminating conflicting constraints. This setting is only relevant in 3D.

You must always enter a value for the sector angle θ_S.

In either case, enter an **Azimuthal mode number** for the mode to be studied. It can vary from 0 to $N/2$, where N is the total number of sectors on a full revolution.

- For **User defined** select the checkbox for any of the displacement or rotation components as needed. Then for each selection, choose the **Type of periodicity** — **Continuity** or **Antiperiodicity**. Each selected displacement component will be connected by $u_i(\mathbf{x}_d) = u_i(\mathbf{x}_s)$ or $u_i(\mathbf{x}_d) = -u_i(\mathbf{x}_s)$. Each selected rotation component will be connected by $a_i(\mathbf{x}_d) = a_i(\mathbf{x}_s)$ or $a_i(\mathbf{x}_d) = -a_i(\mathbf{x}_s)$ If the source and destination objects are rotated with respect to each other, a transformation is performed using the selected coordinate systems so that corresponding components of the degrees of freedom are connected.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

ORIENTATION OF SOURCE

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For information about the **Orientation of Source** section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

ORIENTATION OF DESTINATION

This section appears if the setting for **Transform to intermediate map** in the **Orientation of Source** section is changed from the default value, **Automatic**, and **Advanced Physics Options** is selected in the **Show More Options** dialog. For information about the

Orientation of Destination section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

MAPPING BETWEEN SOURCE AND DESTINATION

For information about the **Mapping Between Source and Destination** section, see Mapping Between Source and Destination in the *COMSOL Multiphysics Reference Manual*.

- Cyclic Symmetry and Floquet Periodic Conditions in the *Structural Mechanics Theory* chapter.
- Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

Vibrations of an Impeller: Application Library path
Structural_Mechanics_Module/Dynamics_and_Vibration/impeller

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Periodic Condition**
Plate > **Periodic Condition**

Ribbon

Physics tab with **Shell** selected:

Edges > **Connections** > **Periodic Condition** (3D)
Points > **Connections** > **Periodic Condition** (2D axisymmetry)

Physics tab with **Plate** selected:

Boundaries > **Periodic Condition**

Adiabatic Heating, Layered

The **Adiabatic Heating, Layered** node adds the equations for adiabatic heating in layered shells caused by abrupt changes in temperature due to fast deformation.

The increase in temperature is then defined by solving the distributed equation

$$\rho C_p \dot{T} = \beta_{ah} Q_h$$

where ρ is the density, C_p is the heat capacity at constant pressure, T is the temperature field, β_{ah} is the coefficient of adiabatic heating, and Q_h corresponds to the heat sources due to mechanical dissipative processes.

Adiabatic heating is only available with some COMSOL products (see www.comsol.com/products/specifications/).

If the Composite Materials Module is available, adiabatic heating can be applied to arbitrary layers. The material properties, orientations, and layer thicknesses are defined using **Layered Material** node. The offset, and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

Without the Composite Materials Module, only single layer shells and membranes can be modeled. This is still useful, for example for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered Materials in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

It is only possible to select boundaries which are part of the selection of a layered material defined in **Layered Material Link** or **Layered Material Stack** node.

SHELL PROPERTIES

Select the layers in which adiabatic heating is to be modeled.

TABLE 6-24: LAYER SELECTION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the same layer is selected in two nodes being active on the same boundary, the contributions will be exclusive

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

INITIAL VALUES

Enter the **Initial temperature** T_{ini}. The default value is 293.15 K.

THERMODYNAMICS

The density ρ is taken from the material model (Linear Elastic Material or Hyperelastic Material).

The default **Heat capacity at constant pressure** C_{p} uses values **From material**. For **User defined**, enter an expression or value. The default value for the **User defined** is `0` J/(kg·K).

Enter the **Coefficient of adiabatic heating**, β_{ah}. The default value is 1 (dimensionless), which means that dissipative processes contribute 100% as heat sources.

Select the **Dissipative heat source** — **Include all dissipative sources** or **User defined**.

The **Dissipative heat source** list makes it possible to include specific heat sources for the adiabatic heating. Enter a value or expression for the heat source Q_{h} to include. For instance, the dissipated energy density due to creep is available under the variable `shell.Wc` and due to viscoplasticity under the variable `shell.Wvp`. Here `solid` denotes the name of the physics interface node.

TIME STEPPING

The **Backward Euler** method is not available with the Layered Shell interface neither with the Layered Linear Elastic Material nor the Layered Hyperelastic Material in the Shell and Membrane interfaces.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of this field can affect the convergence of the overall solution.

LOCATION IN USER INTERFACE

Context Menus

Shell > **More** > **Adiabatic Heating, Layered**

Ribbon

Physics tab with **Shell** selected:

Boundary > **More** > **Adiabatic Heating, Layered**

Boundary to Boundary

Add a **Boundary to Boundary** node to connect two shell boundaries that are geometrically disjoint. It is assumed that the boundaries are parallel to each other. The connection between corresponding points on the two boundaries can be rigid or flexible.

This connection is useful in, for example, the following situations:

- Two plates are joined by a thin glue layer.
- It is physically a single plate, but for modeling reasons you want to extend it on different boundaries in different directions.

The mesh does not have to be the same on both boundaries, but there will be less local stress fluctuations if it is.

When deciding on which boundaries so use as source and destination, there are two considerations:

- If there is a significant difference in mesh size, then the side with finer mesh should be selected as destination.
- It the mesh size is similar, it will me more efficient to use the boundaries with the smaller total area as destination.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the stiffness in the case of a flexible connection. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Shell Local System**
- Any additional user-defined coordinate system.

CONNECTION SETTINGS

Select the **Method** — **Rigid** or **Flexible**.

For **Rigid**, the coupling between the two boundaries is rigid in a pointwise manner. Every node on the destination boundary is coupled by a virtual rigid bar to the nearest point on the source boundary. This does not imply that the whole boundary is rigid, such type of coupling can be obtained using a Rigid Connector.

For **Flexible**, select the **Connection properties** — **From material** or **User defined**.

When data is taken **From material**, the elastic material properties of the **Joint material** are taken from the boundary material on the source side, or other material available in the **Material** node.

The default values of the **Young's modulus** E, **Poisson's ratio** ν, and **Density** ρ are taken **From material**. Select **User defined** to enter other values or expressions.

For **User defined** connection, enter the **Spring constant per reference area**. This is a matrix $k_{u,A}$, coupling the displacements on the two sides.

$$\mathbf{f} = -k_{u,A}\Delta\mathbf{u}$$

Here, $\mathbf{f}$ is the force acting on the destination side, and $\Delta\mathbf{u}$ is the difference of the displacement of the destination and source sides. All vectors are expressed in the Coordinate System Selection.

Optionally, you can enter a **Mass per reference area**, $\rho_{u,A}$. This can for example be the mass of a glue layer.

The sides of the shells (top or bottom) that are to be connected must be known. The information is used for two purposes: to connect to the correct set of displacements, and to compute the distance, including the shell thickness and offset.

Connected Location

Select **Connected location** — **Automatic** or **Manual**. In the automatic method, the two sides of the shells that are found to be closest to each other are connected. If you want to make another choice, or the if the automatic search for some reason does not give the intended result, select **Manual**.

In the manual method, the **Source** and **Destination** lists are shown. Select **Top surface** or **Bottom surface** from both lists.

There are some uncommon cases for which the automatic mode will fail. If this happen, you will get an error message. You can then either move to manual mode or, for some cases, use several **Boundary to Boundary** nodes with smaller selections.

Connection Tolerance

The two boundaries are only connected if they are within a certain tolerance from touching each other. The distance is computed including shell thickness and offset, as well as the selected connection boundaries. You can modify this tolerance.

Select the **Connection tolerance** — **Automatic** or **User defined**. For **User defined**, enter the maximum allowable distance or overlap between the boundaries, δ.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Search method** — **Closest point** or **Manual** to specify what should happen if an evaluation point in the destination boundary is mapped outside the source boundary.

The **Closest point** method is the default and most robust search method. The closest point in the source selection is used by the mapping operator, even if that point is far away from the destination point. This method is not as efficient as the **Manual** method.

The **Manual** method is an alternative search method that is computationally efficient, but its robustness is determined by the curvature of the boundaries, the **Search distance**, and the **Extrapolation tolerance**. Specify the maximum search distance between the source and destination boundaries and the extrapolation tolerance.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. The information in this section i sonly used if **Method** is set to **Rigid**.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Boundary to Boundary** (Boundary)

Ribbon

Physics tab with **Shell** selected:

Edge to Boundary

Add an **Edge to Boundary** node to connect a shell edge to a shell boundary in a mesh independent configuration.

This connection is useful in, for example, the following situations:

- You have an imported geometry where there are slits in the geometry.
- A midsurface generation has left slits in the geometry.
- You want to avoid the extra flexibility that would be the effect of connecting at the midsurface. With this type of coupling, you can take she shell thickness into account in a more accurate way.
- You want to model and evaluate a weld. In that case, you can use the flexible version of the coupling, and evaluate the forces in it.

If possible, use the Edge to Edge coupling, which involves fewer approximations. The **Edge to Boundary** coupling has the advantage that there is no need for having matching edges in the geometry.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the stiffness in the case of a flexible connection. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Shell Local System**
- **Local edge system**

For details about the definition of local edge systems, see Local Edge System.

- Any additional user-defined coordinate system.

FACE DEFINING THE ORIENTATIONS

This setting is used in conjunction with **Local edge system** and **Shell Local System**. When the connection is used for an edge which is shared between boundaries, the coordinate

system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface and is only visible if the selected coordinate system is **Local edge system** or **Shell Local System**.

CONNECTION SETTINGS

Select a **Method** — **Rigid** or **Flexible**.

For **Rigid**, the coupling between the two parts is rigid in a pointwise manner. Every point on the destination boundary is coupled by a virtual rigid bar to the nearest point on the source edge. This does not imply that the whole edge is rigid.

For **Rigid**, select for both the source and destination sides of the connection the **Connected location** — **Top surface**, **Midsurface** (default), or **Bottom surface**.

For **Flexible**, you supply a stiffness matrix, connecting the corresponding points on the source and destination sides. The stiffness coefficients are given per reference length along the edge. Input fields for the matrices $\mathbf{k}_{u,L}$ and $\mathbf{k}_{\Theta,L}$ are always shown. Select **Translational-rotational coupling** to show input fields for the coupling matrices $\mathbf{k}_{u\Theta,L}$ and $\mathbf{k}_{\Theta u,L}$. In 2D axisymmetry, most elements of these matrices are by definition zero. Only elements which can be nonzero are shown; these are elements 13 and 23 for $\mathbf{k}_{u\Theta,L}$, and elements 31 and 32 for $\mathbf{k}_{\Theta u,L}$.

The stiffness relation can be expressed as

$$\begin{bmatrix}\mathbf{f}\\ \mathbf{m}\end{bmatrix} = -\begin{bmatrix}\mathbf{k}_{u,L} & \mathbf{k}_{u\Theta,L}\\ \mathbf{k}_{\Theta u,L} & \mathbf{k}_{\Theta,L}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{u}\\ \Delta\Theta\end{bmatrix}$$

Here, $\mathbf{f}$ and $\mathbf{m}$ are the forces and moments per reference length acting on the destination side, and $\Delta\mathbf{u}$ is the difference between the destination side displacements and the source side displacements. Similarly, $\Delta\Theta$ is the difference between the destination side rotations and the source side rotations. All vectors are expressed in the selected coordinate system.

Optionally, you can enter a **Mass per reference length**, $\rho_{u,L}$. This can for example be the mass of a not modeled weld.

On the boundary, a strip with a certain width is used for the connection. Select a **Connected area defined by** — **Shell thickness** or **Distance from shell midsurface** to control

the width of the strip. The default is **Shell thickness**, in which case the thickness of the shell on the source side (the edge) is used as the connection width. By choosing **Distance from shell midsurface** enter the desired width.

The two boundaries are only connected if they are within a certain tolerance from touching each other. The distance is computed including shell thickness and offset. You can modify this tolerance. Select a **Connection tolerance** — **Automatic** or **User defined**. For **User defined**, enter the maximum allowable distance or overlap between the boundaries, δ.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. The information in this section i sonly used if **Method** is set to **Rigid**.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Edge to Boundary** (Edge)

Ribbon

Physics tab with **Shell** selected:

Edges > **Connections** > **Edge to Boundary**

Edge to Edge

Add an **Edge to Edge** node to connect two shell edges that are geometrically disjoint. It is assumed that the edges are parallel. The connection between corresponding points on the two edges can be rigid or flexible.

This connection is useful in, for example, the following situations:

- You have an imported geometry where there are slits between edges.
- A midsurface generation has left slits between edges.

- You want to avoid the extra flexibility that would be the effect of connecting at the midsurface using a common edge. With this type of coupling, you can take she shell thickness into account in a more accurate way.
- You want to model and evaluate a weld. In that case, you can use the flexible version of the coupling, and evaluate the forces in it.

If there is no good match between two edges, you can as an alternative use the Edge to Boundary coupling.

When one or more **Edge to Edge** nodes are active in the model, a **Connected Region Indicator** plot group will be added to the list of result templates. It contains a line plot showing the parts of the destination edges where connection is active. If there are other connection features in the Shell interface, all connection indicators appear in the same plot group.

COORDINATE SYSTEM SELECTION

Select the coordinate system to use for specifying the stiffness in the case of a flexible connection. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).
- **Shell Local System**
- **Local edge system**

For details about the definition of local edge systems, see Local Edge System.

- Any additional user-defined coordinate system.

FACE DEFINING THE ORIENTATIONS

This setting is used in conjunction with **Local edge system** and **Shell Local System**. When the connection is used for an edge which is shared between boundaries, the coordinate system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

This section is available only for edges in the Shell interface and is only visible if the selected coordinate system is **Local edge system** or **Shell Local System**.

CONNECTION SETTINGS

Select a **Method** — **Rigid** or **Flexible**.

For **Rigid**, the coupling between the two edges is rigid in a pointwise manner. Every node on the destination edge is coupled by a virtual rigid bar to the nearest point on the source edge. This does not imply that the whole edge is rigid. Such a coupling can be obtained using a Rigid Connector.

For **Rigid**, select for both the source and destination sides of the connection the **Connected location** — **Top surface**, **Midsurface** (default), or **Bottom surface**.

For **Flexible**, you supply a stiffness matrix, connecting the corresponding points on the source and destination edges. The stiffness coefficients are given per reference length along the edge. Input fields for the matrices $\mathbf{k}_{u,L}$ and $\mathbf{k}_{\Theta,L}$ are always shown.

Select **Translational-rotational coupling** to show input fields for the coupling matrices $\mathbf{k}_{u\Theta,L}$ and $\mathbf{k}_{\Theta u,L}$. In 2D axisymmetry, most elements of these matrices are by definition zero. Only elements which can be nonzero are shown; these are elements 13 and 23 for $\mathbf{k}_{u\Theta,L}$, and elements 31 and 32 for $\mathbf{k}_{\Theta u,L}$.

The stiffness relation can be expressed as

$$\begin{bmatrix}\mathbf{f}\\ \mathbf{m}\end{bmatrix} = -\begin{bmatrix}\mathbf{k}_{u,L} & \mathbf{k}_{u\Theta,L}\\ \mathbf{k}_{\Theta u,L} & \mathbf{k}_{\Theta,L}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{u}\\ \Delta\Theta\end{bmatrix}$$

Here, $\mathbf{f}$ and $\mathbf{m}$ are the forces and moments per reference length acting on the destination side, and $\Delta\mathbf{u}$ is the difference between the destination side displacements and the source side displacements. Similarly, $\Delta\Theta$ is the difference between the destination side rotations and the source side rotations. All vectors are expressed in the selected coordinate system.

Optionally, you can enter a **Mass per reference length**, $\rho_{u,L}$. This can for example be the mass of a not modeled weld.

In order to ascertain that the two edges are parallel, a certain tolerance is used when comparing the angles between them. You can modify this tolerance. Select the **Connection tolerance** — **Automatic** or **User defined**. For, Δ, enter the maximum angle between the edges.

Select the **Weld verification** checkbox to compute forces and stresses per reference length and their average along the weld.

Select the **Weld type** — **Double-sided fillet**, **Single-sided fillet (top)**, **Single-sided fillet (bot)**, or **Butt**. For any of the options, enter the **Throat size** a, the **Allowable equivalent stress** σ_{eq}^{max}, and the **Allowable normal stress (perpendicular to the weld)** $\sigma_{\perp}^{max}$.

Evaluating Stresses in Continuous Welds

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. The information in this section is only used if **Method** is set to **Rigid**.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Edge to Edge**

Ribbon

Physics tab with **Shell** selected:

Edges > **Connections** > **Edge to Edge**

Spot Welds

Add one or more **Spot Welds** nodes to model sets of spot welds. A single **Spot Welds** node can accommodate any number of spot welds, as long as they share the same properties. The spot weld locations are entered as a list of coordinate triplets, and are independent of geometrical points or node points.

The connection is elastic. The spring stiffness can either be viewed as a pure penalty formulation of a constraint, or be directly related to the physical flexibility of the

connection. Since the connection is distributed across the nugget area, the sport weld also possess bending and torsional stiffness.

When one or more **Spot Welds** nodes are active in the model, six entries are added to the list of result templates:

- A **Spot Welds Forces** plot group, to show the normal and shear forces and a label showing the identity number of each weld. Evaluate the **Spot Welds Normal Forces** and **Spot Welds Shear Forces** evaluation groups to display the spot welds forces.
- A **Spot Welds Normal Forces** evaluation group that contains a table of the location and the normal forces of all welds. Evaluate this table to display the forces in the **Spot Welds Forces** plot group.
- A **Spot Welds Shear Forces** evaluation group that contains a table of the location and the shear forces of all welds. Evaluate this table to display the forces in the **Spot Welds Forces** plot group.
- A **Spot Welds Forces Magnitude** evaluation group that contains a table of the normal and shear forces magnitude of all welds.
- A **Nugget Location** plot group that contains a surface plot showing the nugget locations, a point plot at Gauss points showing the location of the nuggets on the destination boundary, and a mesh plot.
- A **Connected Region Indicator** plot group that contains a surface plot showing the areas on the destination surface where the spot welds are active. If there are other connection features in the Shell interface, all connection indicators appear in the same plot.

You must delete the **Spot Weld Forces** plots and evaluation groups and add them again from the list of result templates if you change the number of spot welds in your model or compute new results.

BOUNDARY SELECTION

Select the boundaries that form one side of the set of spot welds.

BOUNDARY SELECTION, DESTINATION

Select the boundaries that form the other side of the set of spot welds.

The equations for the connection are set up on the destination side. If there is a difference in mesh density, select the side with the finer mesh as destination.

	Although spot welds are independent of geometrical points or node points and the connection is distributed across the nugget area, achieving accurate results requires that the element size on the destination side not be significantly larger than the nugget diameter.

NUGGET LOCATION

In the table, enter the coordinates of the centers of the spot welds. The coordinates do not have to be exactly on the shell surface. Each given spot weld location will be projected onto the nearest point on the destination boundary.

- Use the **Add** button ( ) to add a row to the table, the **Delete** button ( ) to delete a row in the table, or the **Clear Table** button ( ) to clear the whole table.
- Use the **Load from file** button ( ) and the **Save to file** button ( ) to load and store the list of nugget locations using a text file with space-separated columns.

The sides of the shells (top or bottom) that are to be connected must be known. The information is used for two purposes: to connect to the correct set of displacements, and to compute the distance, including the shell thickness and offset.

Select **Connected location** — **Automatic** or **Manual**. In the automatic mode, the two sides of the shells that are found to be closest to each other are connected. If you want to make another choice, or the if the automatic search for some reason does not give the intended result, select **Manual**.

In manual mode, the **Source** and **Destination** lists are shown. Select **Top surface** or **Bottom surface** from both lists.

	There are some uncommon cases for which the automatic mode will fail. If this happen, you will get an error message. You can then either move to manual mode or, for some cases, use several **Spot Welds** nodes having smaller selections.

NUGGET PROPERTIES

Enter the **Nugget diameter**, d_{nug}.

Select the **Nugget stiffness** — **From material** or **User defined**.

When data is taken **From material**, the elastic material properties (E, ν) on the destination side are used.

Enter the **Cohesive thickness**, L_{cohe}. This is a distance in the direction perpendicular to the plate used to compute the stiffness of the connection. Its value should be of the order of the spacing between the welded plates. The default value is one percent of the nugget diameter, $d_{\text{nug}}/100$.

The normal and shear stiffnesses (per reference area) are then computed as

$$k_{\text{n}} = \frac{1}{L_{\text{cohe}}} \cdot \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$$

$$k_{\text{s}} = \frac{1}{L_{\text{cohe}}} \cdot \frac{E}{2(1+\nu)}$$

The total stiffness of a spot weld is obtained by multiplying by the cross-section area of the nugget, as defined by its diameter.

For **User defined** nugget stiffness, enter values or expressions for the **Normal stiffness**, k_{n}, and **Shear stiffness**, k_{s}. These are the total stiffnesses of a single spot weld (SI unit N/m).

QUADRATURE SETTINGS

The stiffness that each spot weld generates between the two connected boundaries is computed using numerical integration. If the size of the mesh element is large relative to the nugget diameter, a high integration order may be needed to get an accurate estimate of the integral. Using very high integration orders together with a large number of spot welds can have a negative impact on the solution time.

When **Use automatic quadrature settings** is selected, the integration order is set to 8, which means that there are 5-by-5 integration points in a quadrilateral element.

Clear **Use automatic quadrature settings** to manually enter an integration order. The maximum value is 30, which means 16-by-16 points in a quadrilateral element.

!	Use the Nugget Location plot group available in the Result Templates window to visualize the location of the integration point. In the Point plot settings, under the Evaluation section, you can try a different Gauss-point order.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Search method** — **Closest point** or **Manual** to specify what should happen if an evaluation point in the destination boundary is mapped outside the source boundary.

The **Closest point** method is the default and most robust search method. The closest point in the source selection is used by the mapping operator, even if that point is far away from the destination point. This method is not as efficient as the **Manual** method.

The **Manual** method is an alternative search method that is computationally efficient, but its robustness is determined by the curvature of the boundaries, the **Search distance**, and the **Extrapolation tolerance**. Specify the maximum search distance between the source and destination boundaries and the extrapolation tolerance.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

- Modeling Spot Welds
- Theory for Spot Welds

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Spot Welds**

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Connections** > **Spot Welds**

Fasteners

The **Fasteners** node is intended for a simplified modeling of fasteners like rivets or bolts. It is based on the assumption that the holes for the fasteners are present in the shell

geometry. The two plates are joined using an elastic connection. There are two possible types of connections:

- The edges of matching holes are connected.
- Boundaries surrounding the matching holes are connected. Such boundaries should typically match the size of rivet heads or washers.

The latter type of connection is somewhat more accurate, and can reduce local stress concentrations. This is in particular true when bolts are modeled. Making such connections requires a more sophisticated representation of the geometry, where the fastener holes are surrounded by suitable, usually annular, boundaries.

The actual selection of the pairs of holes to be connected is done in a Hole Selection subnode.

When one or more **Fasteners** nodes are active in the model, five entries are added to the list of result templates:

- A **Fasteners Forces** plot group that shows the normal and shear forces and a label showing the identity number of each fastener. Evaluate the **Fasteners Normal Forces** and **Fasteners Shear Forces** evaluation group to display the fasteners forces. The coloring can be affected by the addition of a **Safety** subnode.
- A **Fasteners Normal Forces** evaluation group that contains a table of the location, normal forces, and damage index of all fasteners. Evaluate this table to display the forces in the **Fasteners Forces** plot group.
- A **Fasteners Shear Forces** evaluation group that contains a table of the location, shear forces, and damage index of all fasteners. Evaluate this table to display the forces in the **Fasteners Forces** plot group.
- A **Fasteners Forces Magnitude** evaluation group that contains a table of the normal and shear forces magnitude for all fasteners.
- A **Connected Region Indicator** plot group that contains a line plot or a surface plot showing the hole edges or hole surrounding boundaries, respectively, on the destination surface where the fasteners are active. If there are other connection features in the Shell interface, all connection indicators appear in the same plots.

You must evaluate the **Fasteners Normal Forces** and **Fasteners Shear Forces** evaluation groups to update the **Fasteners Forces** plot group if you change the number of fasteners in your model or compute new results. Also, remove the **Connected Region Indicator** plot group and replace it by a new one if you change the hole selection type between edge or boundary.

LOCATION

The sides of the shells (top or bottom) that are to be connected must be known. The information is used for two purposes: to connect to the correct set of displacements, and to compute the distance, including the shell thickness and offset. Select **Connected location** — **Automatic** or **Manual**. In the automatic mode, the two sides of the shells that are found to be closest to each other are connected. If you want to make

another choice, or the if the automatic search for some reason does not give the intended result, select **Manual**.

In manual mode, the **Source** and **Destination** lists are shown. Select **Top surface** or **Bottom surface** from both lists.

There are some uncommon cases for which the automatic mode will fail. If this happen, you will get an error message. You can then either move to manual mode or, for some cases, use several **Fasteners** nodes having smaller selections.

FASTENER PROPERTIES

Enter **Diameter**, d_f for the fasteners diameter.

The stiffness of the fasteners can be computed in two ways, either using analytical expressions based on that the fastener is approximated as a beam, or by direct entry of stiffness values. From the **Fastener model** list, select **Beam** or **Spring**.

Beam

The normal stiffness, k_n, and shear stiffness, k_s, of the fastener are computed using the following expressions:

$$k_n = \frac{\pi E_f d_f^2}{4(t_{dst} + t_{src})}$$

$$k_s = \frac{1}{C_b + C_s + C_{br} + C_{pbr}}$$

The four compliance contributions are defined as

$$C_b = \alpha_b \frac{3t_{src}^3 + 16t_{src}^2 d_{dst} + 16t_{dst}^2 d_{src} + 3t_{dst}^3}{2\pi E_f d_f^4}$$

$$C_s = \alpha_s \frac{7 + 14\nu_f + 8\nu_f^2}{2(1+\nu_f)} \cdot \frac{t_{dst} + t_{src}}{\pi E_f d_f^2}$$

$$C_{br} = \alpha_{br}\left(\frac{1}{t_{src}E_f} + \frac{1}{t_{dst}E_f}\right)$$

$$C_{pbr} = \alpha_{pbr}\left(\frac{1}{t_{src}E_{src}} + \frac{1}{t_{dst}E_{dst}}\right)$$

In these expressions

- d_f is the diameter of the fastener.
- E_f, E_{src}, and E_{dst} are the elastic moduli of the fastener, the shell on the source side, and the shell on the destination side respectively.
- ν_f is the Poisson's ratio of the fastener.
- t_{src} and t_{dst} are the thicknesses of the shell on the source and destination sides respectively.
- The multipliers α_m are multipliers that make it possible to adjust the default analytical values.

Enter the following data for the fasteners:

- **Young's modulus**, E_f.
- **Poisson's ratio**, ν_f.
- **Bending flexibility multiplier**, α_b.
- **Shear flexibility multiplier**, α_s.
- **Fastener bearing flexibility multiplier**, α_{br}.
- **Plate bearing flexibility multiplier**, α_{pbr}.

Spring

Enter the **Normal stiffness**, k_n, and **Tangential stiffness**, k_s.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Search method** — **Closest point** or **Manual** to specify what should happen if an evaluation point in the destination boundary is mapped outside the source boundary.

The **Closest point** method is the default and most robust search method. The closest point in the source selection is used by the mapping operator, even if that point is far away from the destination point. This method is not as efficient as the **Manual** method.

The **Manual** method is an alternative search method that is computationally efficient, but its robustness is determined by the curvature of the boundaries, the **Search**

distance, and the **Extrapolation tolerance**. Specify the maximum search distance between the source and destination boundaries and the extrapolation tolerance.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

- Simplified Modeling of Fasteners
- Theory for Fasteners

LOCATION IN USER INTERFACE

Context Menus
Shell > **Fasteners**

Ribbon
Physics tab with **Shell** selected:

Global > **Shell** > **Fasteners**

Hole Selection

The **Hole Selection** node is used to specify pairs of holes that will be connected by fasteners. The connection between the two shell boundaries can be defined either by referencing the hole edges directly or by using the boundaries surrounding the holes. Each setup includes at least one **Hole Selection** subnode, and additional subnodes can be added as needed.

A single **Hole Selection** node is usually sufficient for the default automatic search for matching hole pairs. When using manual selections, you need to add one **Hole Selection** node per hole pair.

EDGE SELECTION, SOURCE

This section is enable when **Source selection type** is set to **Edge**.

The way the edge selections are used depends on whether automatic or manual edge selection is used.

When an **Edge Selection** node is added, the selection defaults to **All edges**, which implies that all edges on the source selection in the parent **Fasteners** node are shown as

applicable. This is the standard configuration when working in the automatic mode. You can, however, remove some edges from the list.

When working in manual mode, select a set of edges that define the hole on the source side.

In either case, all edges around a hole must be selected in order for a hole to be defined. The **Group by continuous tangent** option is a useful tool for making fast and correct manual selections of holes.

EDGE SELECTION, DESTINATION

This section is enable when **Destination selection type** is set to **Edge**.

The selection of destination side hole edges works the same way as described above for the source side selection.

BOUNDARY SELECTION, SOURCE

This section is enable when **Source selection type** is set to **Boundary**.

The way the boundary selections are used depends on whether automatic or manual edge selection is used.

When an **Hole Selection** node is added, the selection defaults to **All boundaries**, which implies that all boundaries on the source selection in the parent **Fasteners** node are shown as applicable. This is the standard configuration when working in the automatic mode. You can, however, remove some boundaries from the list to avoid that they are taken into account.

When working in manual mode, select a set of boundaries that surround the hole on the source side.

In either case, all boundaries around a hole must be selected in order for a hole to be defined.

BOUNDARY SELECTION, DESTINATION

This section is enable when **Destination selection type** is set to **Boundary**.

The selection of destination side hole boundaries works in the same way as described in Boundary Selection, Source.

SELECTION TYPE

From the **Source selection type** list, select **Edge** or **Boundary**. This defines the type of connection that is made on the source shell boundaries.

From the **Destination selection type** list, select **Edge** or **Boundary**. This defines the type of connection that is made on the destination shell boundaries.

In Shell interface, **Edge** is set as default selection type for both source and destination selection.

FASTENER HOLE DETECTION

Select a method for detecting the fastener holes, **Automatic** or **Manual**. For **Automatic**, adjust **Diameter tolerance**, **Offset tolerance**, **Maximum angle**, and **Annular center offset**. These values control when two holes are connected to form a pair. The last parameter is available only when the selection type for either source or destination is set to boundary and the checkbox **Search annular shape only** is selected. Selecting this checkbox adds the requirement that the region around a hole is actually annular. When deselected, any boundary around a hole is valid as the connecting boundary.

The exact definitions are given in Automatic Hole Detection in the theory chapter.

In the automatic mode, the source and destination boundaries selected in the parent Fasteners node are scanned in order to find all holes. The scan is limited to the active selections. Hole pairs are connected when holes on both the source and destination have the diameters within a given tolerance with respect to the fastener diameter, and have their centers located on the same axis normal to the two boundaries.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Fasteners** > **Hole Selection**

Ribbon

Physics tab with **Fasteners** node selected in the model tree:

Attributes > **Hole Selection**

Rigid Connector

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. A rigid connector can connect an arbitrary combination of edges and points which all will move together as being attached to a virtual rigid object.

You can add the **Rigid Connector** node at the boundary, edge, and point levels.

- When added at the boundary level, you can connect boundaries, edges, and points as long as at least one boundary is selected. Selecting edges and points is optional.
- When added at the edge level, you can connect edges and points as long as at least one edge is selected. Selecting points is optional.
- When added at the point level, you can connect a set of points to each other.

If your selection is too small, you can introduce rigid body motions in the model. No exact rules can be given because there are many possible configurations where, for example, boundaries are connected to each other or affected by other constraints. Also, you can, by providing proper constraints for the rigid connector, suppress rigid body motions.

When the rigid connector is added at point level, a set of consistency checks is performed. You can suppress these checks by clearing the **Include consistency checks** checkbox in the **Advanced** section.

If the study step is geometrically nonlinear, the rigid connector takes finite rotations into account.

Rigid connectors are available in the Solid Mechanics, Multibody Dynamics, Shell, Beam, and Pipe Mechanics interfaces. Rigid connectors from different interfaces can be attached to each other.

You can add functionality to the rigid connector through the following subnodes:

- Applied Force (Rigid Connector) to apply a force in given point.
- Applied Moment (Rigid Connector) to apply a moment.
- Mass and Moment of Inertia (Rigid Connector) to add extra mass and moment of inertia in a given point.

Spring Foundation (Rigid Connector) to add a translational or rotational spring or damper in a given point.When physics symbols are shown, a rigid connector is

represented by a symbol at the center of rotation, together with a set of lines connecting the center of rotation with the centers of gravity of the selected entities.

- When the rigid connector is added at the boundary level, such symbols are shown only for the selected boundaries, but not for auxiliary selections of edges or points.
- When the rigid connector is added at the edge level, such symbols are shown only for the selected edges, but not for auxiliary selections of points.
- Because of the way physics symbols are evaluated, as a lightweight operation when moving between physics nodes in the model builder tree, it is sometimes not possible to determine the center of rotation. In particular, if an offset is supplied, it will not be taken into account.

BOUNDARY SELECTION

This section is present when the **Rigid Connector** node has been added at the boundary level. Select one or more boundaries to be part of the rigid region.

EDGE SELECTION

This section is present when the **Rigid Connector** node has been added at the boundary or edge level.

- When the **Rigid Connector** is added at the edge level, select one or more edges that form part of the rigid region.
- When the **Rigid Connector** is added at the boundary level, this section is initially collapsed. Here, you can add optional edges to the rigid region. The edges cannot be adjacent to the selected boundaries.

POINT SELECTION

This section is always present.

- When the **Rigid Connector** is added at the point level, select a number of points that form the rigid region.
- When the **Rigid Connector** is added at the boundary or edge levels, this section is initially collapsed. Here, you can add optional points to the rigid region. The points cannot be adjacent to the selected boundaries or edges.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed displacements or rotations are specified along the axes of this coordinate system. It is also used for defining the axis directions of the moment of inertia tensor of the **Mass and Moment of Inertia** subnode.

In order to be applicable, a coordinate system must have axis directions that are independent of the location in space.

CENTER OF ROTATION

The center of rotation serves two purposes.

- If you prescribe the displacement of the rigid connector, this is the place where it is fixed.
- Results are interpreted with respect to the center of rotation.

Select a **Center of rotation** — **Automatic**, **Centroid of selected entities**, or **User defined**.

- For **Automatic** the center of rotation is at the geometrical center of the selected edges. The constraints are applied at the center of rotation.
- For **Centroid of selected entities**, a subnode for selection of the entities is added to the Model Builder. The center of rotation is located at the centroid of the selected entities, which do not need to be related to the selection of the rigid connector.

When points are selected, It is the geometrical location of the points that is used for computing the centroid. Any offset of the shell at the points is ignored.

- When the rigid connector is added at boundary or edge level, select an **Entity level** — **Edge** or **Point**.
- When the rigid connector is added at point level, the centroid can only be defined by a point selection. As a special case, you can select a single point, and thus locate the center of rotation at a certain point.

- For **User defined**, in the **Global coordinates of center of rotation** X_C table enter coordinates based on space dimension.

- For a rigid connector at edge level, when **Centroid of selected entities** is chosen, a default Center of Rotation: Edge or Center of Rotation: Point subnode is added, depending on the setting of **Entity level**.
- For a rigid connector at point level, when **Centroid of selected entities** is chosen, a default Center of Rotation: Point subnode is added.
- The center of rotation is located at the centroid of the selected entities, which do not need to be related to the rigid connector itself. As a special case, you can select a single point, and thus use that point as center of rotation.
- When a Center of Rotation: Point node is used for selection, any point in the geometry can be selected, even if it is not part of the physics interface.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_C = \mathbf{X}_{C,\text{input}} + \mathbf{X}_{\text{offset}}$$

PRESCRIBED DISPLACEMENT AT CENTER OF ROTATION

To define a prescribed displacement for each spatial direction x, y, and z select one or all of the **Prescribed in x**, **Prescribed in x**, and **Prescribed in z direction** checkboxes. Then enter a value or expression for the prescribed displacements u_{0x}, u_{0y}, or u_{0z}.

PRESCRIBED ROTATION

Select an option from the **By** list — **Free** (the default), **Constrained rotation**, or **Prescribed rotation**.

- For **Constrained rotation** select one or more of the **Constrain rotation about x**, **Constrain rotation about y**, and **Constrain rotation about z axis** checkboxes in order to enforce zero rotation about the corresponding axis in the selected coordinate system.
- For **Prescribed rotation** enter an **Axis of rotation** Ω and an **Angle of rotation** ϕ_0. The axis of rotation is given in the selected coordinate system.

RELEASED DEGREES OF FREEDOM

In some cases it can be useful to not constrain the displacement in a certain direction. For instance, the radial displacement of a Rigid Connector acting on a circular cross-section could be allowed to be free. To do so, select a local **Coordinate system** for specifying the directions in which the degree of freedom will be released. The **Coordinate system** list contains only applicable coordinate systems in the model.

Select one or more of the available checkboxes to release the displacement in the corresponding axis in the selected coordinate system:

- **Release displacement in x1 direction**
- **Release displacement in x2 direction**
- *For 3D components:* **Release displacement in x3 direction**

Note that the **Rigid Connector** introduces its own global displacement degrees of freedom (DOFs) and global rotation DOFs. When certain displacement field components are released, this reduces the number of equations used to solve for the rigid connector's global DOFs, while preserving the number of said DOFs. Therefore, it may become necessary to constrain some of the Rigid Connector's global DOFs to achieve static determinacy.

The section **Released Degrees of Freedom** is only shown if the checkbox **Use weak constraints for rigid-flexible connection** in the section **Constrain Settings** is not enabled.

REACTION FORCE SETTINGS

Select **Evaluate reaction forces** to compute the reaction force caused by a prescribed motion. The default is to not compute the reaction force. When selected, the prescribed motion is implemented as a weak constraint.

Select **Apply reaction only on rigid body variables** to use a unidirectional constraint for enforcing a prescribed motion. The default is that bidirectional constraints are used. This setting is useful in a situation where a bidirectional constraint would give an unwanted coupling in the equations. This would happen if the prescribed value of the motion is a variable solved for in other equations.

FORMULATION

Some contributions from a rigid connector will, under geometric nonlinearity, result in a nonsymmetric local stiffness matrix. If all other aspects of the model are such that the global stiffness matrix would be symmetric, then such a nonsymmetric contribution may have a heavy impact on the total solution time and memory usage. In such cases, it is often more efficient to use an approximative local stiffness matrix that is symmetric.

Select **Use symmetric formulation for geometric nonlinearity** to force all matrix contributions from the rigid connector and its subnodes to be symmetric.

Using an approximative stiffness matrix will in general require more iterations. However, since the computational cost per iteration will be cut at least by a factor of two if a symmetric matrix can be used, it is usually more efficient to ignore a weak nonsymmetry.

In particular, if the rotation of the rigid connector per time step or parameter increment is small, there will in general be no increase in the number of iterations at all if this option is used.

When the global stiffness matrix is nonsymmetric for other reasons, then there is nothing to be gained from symmetrizing the contribution from the rigid connector.

CONSTRAINT SETTINGS

On the edges where the rigid connector is coupled to a flexible material, all nodes on such an edge are constrained to move as a rigid body. As a default, these constraints are implemented as pointwise constraints. If you want to use a weak constraint formulation, select **Use weak constraints for rigid-flexible connection**.

ADVANCED

It is possible to couple rigid connectors to each other. In the **Connect to** list, you can select any rigid connector defined in the Solid Mechanics or Multibody Dynamics interfaces as being rigidly connected to the current one.

When coupling rigid connectors to each other, you must be careful not to add conflicting settings. Typically, you should only assign constraints to one of the connectors, and it is recommended to use a common center of rotation. It is, however, possible to deviate from these recommendations and create other meaningful combinations.

When the rigid connector is added at the point level, the section contains the additional **Include consistency checks** checkbox. The checkbox, which is selected by default, controls whether to perform singularity checks for the selections. If the checks give false positives, you can turn them off by clearing the checkbox. This could, for example, be necessary if a rigid connector, which in itself is singular, is connected to another one in a way that forms a stable configuration.

When the **Rigid Connector** is added at the point level, select the **Add rotational stiffness for two-point selection** checkbox to automatically suppress singularities when the **Include consistency checks** checkbox is disabled. This checkbox allows to enter a rotational stiffness, k_θ, to stabilize the rigid connector. The rotational stiffness is only added if less than three points are selected.

Select **Group dependent variables in solver** — **From physics interface** (default), **Yes**, or **No**, to choose how to group the dependent variables added by the **Rigid Connector** feature in the solver sequence.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can activate and deactivate the rigid connector by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.
- You can assign the value of the prescribed displacement and rotation to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

- Rigid Connector
- Harmonic Perturbation
- Load Cases

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Rigid Connector** (Boundary)
Shell > **Connections** > **Rigid Connector** (Edge)
Shell > **Points** > **Rigid Connector** (Point)

Ribbon

Physics tab with **Shell** selected:

Boundaries > **Connections** > **Rigid Connector**
Edges > **Connections** > **Rigid Connector**
Points > **Shell** > **Rigid Connector**

Thermal Expansion (Rigid Connector)

Add the **Thermal Expansion** subnode to prescribe a deformation of the rigid connector caused by changes in temperature. This makes it possible to reduce stresses caused by the rigid connector being rigid, while there are thermal deformations in the flexible shell to which it is attached.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain in the rigid connector.

Select **Inherit from boundary** to take the thermal expansion data from the boundary to which it is attached. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual material in the rigid connector has the same thermal expansion as the domain itself.

When **Inherit from boundary** is not selected, enter:

- A value or expression for the **Volume reference temperature** T_{ref} that is the temperature at which there are no thermal displacements in the rigid connector.

- A value or expression for **Temperature** T, specifying the temperature distribution of the rigid connector. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The default values **From material** are used. This requires that a material has been assigned to the boundaries of the rigid connector selection. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

- Thermal Expansion Properties.
- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Rigid connector** > **Thermal Expansion**

Ribbon

Physics tab with **Rigid connector** node selected in the model tree:

Attributes > **Thermal Expansion**

Attachment

The **Attachment** node is used to define a set of edges on a flexible shell which can be used to connect it with other components through a joint in the Multibody Dynamics interface. All the selected edges behave as if they were connected by a common rigid body.

Attachments can be added to boundaries in a Multibody Dynamics or a Solid Mechanics interface, to edges in a Shell or Layered Shell interface, or to points in a

Beam interface. This makes it possible to use a joint in the Multibody Dynamics interface for connecting parts modeled in different physics interfaces.

	All other settings are described in the documentation for Attachment in the Solid Mechanics interface.

	The use of attachments is discussed in the Attachments section in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Connections** > **Attachment**

Ribbon

Physics tab with **Shell** selected:

Edges > **Connections** > **Attachment**

Thermal Expansion (Attachment)

Add the **Thermal Expansion** subnode to prescribe a deformation of the attachment caused by changes in temperature. This makes it possible to reduce stresses caused by the attachment being rigid, while there are thermal deformations in the flexible shell to which it is attached.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain in the attachment.

Select **Inherit from boundary** to take the thermal expansion data from the boundary to which it is attached. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual material in the attachment has the same thermal expansion as the boundary itself.

When **Inherit from boundary** is not selected, enter:

- A value or expression for the **Volume reference Temperature** T_{ref} which is the temperature at which there are no thermal displacements in the attachment
- A value or expression for **Temperature** T, specifying the temperature distribution of the attachment. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- Select the **Input type** — **Secant coefficient of thermal expansion**, **Tangent coefficient of thermal expansion**, or **Thermal strain** to specify how the thermal strain is entered. The default values **From material** are used. This requires that a material has been assigned to the boundaries of the attachment selection. When a nonisotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selected in the **Coordinate System Selection** section.

- Thermal Expansion Properties.
- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics > **Attachment** > **Thermal Expansion**
Multibody Dynamics > **Attachment** > **Thermal Expansion**

Ribbon

Physics tab with **Attachment** node selected in the model tree:

Attributes > **Thermal Expansion**

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response, the physics interface splits the harmonic load into two parameters:

- The amplitude, F, which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f:

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

PHASE

Add the phase angle $\mathbf{F}_{\text{ph}}$ for harmonic loads. Enter the phase for each component of the load in the corresponding fields.

MOMENT LOAD PHASE

Add the phase for the moment load $\mathbf{M}_{\text{ph}}$ for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Body Load** > **Phase**
Shell > **Face Load** > **Phase**
Shell > **Edge Load** > **Phase**
Shell > **Point Load** > **Phase**
Shell > **Rigid Connector** > **Applied Force** > **Phase**
Shell > **Rigid Connector** > **Applied Moment** > **Phase**

Ribbon

Physics tab with **Body Load**, **Face Load**, **Edge Load**, **Point Load**, **Applied Force**, or **Applied Moment** selected:

Attributes > **Phase**

Harmonic Perturbation

Use the **Harmonic Perturbation** subnode to specify the harmonic part of nonzero prescribed displacements. This node is used if the study step is frequency response of a perturbation type.

The settings are the same as in the parent Prescribed Displacement/Rotation, Prescribed Velocity, Prescribed Acceleration, or Rigid Connector node. Only degrees of freedom selected as prescribed in the parent node can be assigned a value.

For Prescribed Displacement/Rotation a rotational harmonic part can only be assigned if in the parent node the rotation type (defined in the **By** list) is set to **Rotation**.

See Harmonic Perturbation in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Shell > **Prescribed Velocity** > **Harmonic Perturbation**
Shell > **Prescribed Acceleration** > **Harmonic Perturbation**
Shell > **Rigid Connector** > **Harmonic Perturbation**
Plate > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Plate > **Prescribed Velocity** > **Harmonic Perturbation**
Plate > **Prescribed Acceleration** > **Harmonic Perturbation**

Ribbon

Physics tab with **Prescribed Displacement/Rotation**, **Prescribed Velocity**, **Prescribed Acceleration**, or **Rigid Connector** node selected in the model tree:

Attributes > **Harmonic Perturbation**

7

Layered Shell

This chapter describes the Layered Shell interface ( ), which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Layered Shell Interface

The **Layered Shell (lshell)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used to model layered structural shells on 3D boundaries. Shells are thin flat or curved structures, having significant bending stiffness.

The Layered Shell interface is applicable for thick and moderately thin shells. The formulation resembles that of a stack of fully 3D solid mechanics models, so all stress components, including interlaminar shear stresses can be resolved. For very thin shells this formulation tends to become numerically ill-behaved, and it is then better to use the Shell interface with the **Linear Elastic Material, Layered** model.

The Linear Elastic Material is the default material model. With the Nonlinear Structural Materials Module, the physics interface is extended with more material models, like hyperelasticity, plasticity, creep, and viscoplasticity.

When this interface is added, these default nodes are also added to the **Model Builder** — **Linear Elastic Material**, **Free** (a boundary condition where edges are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **Layered Shell** to select physics features from the context menu.

The Layered Shell interface is only available with the Composite Materials Module (see www.comsol.com/products/specifications/).

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `lshell`.

SHELL PROPERTIES

Here you select on which layers in a layered material that the physics interface should be active. By default, the **Use all layers** checkbox is selected. This means that all layers in all layered materials on the selected boundaries are used.

If you deselect the **Use all layers** checkbox, you can select individual layers within a single layered material. This is a seldom used option, since it means that the physics interface is restricted to the boundaries on which a specific layered material is defined.

For a general description of layer selections, see The Shell Properties and Interface Selection Sections.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasistatic (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

For problems with creep, and sometimes viscoelasticity, the problem can be considered as quasistatic. This is also the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

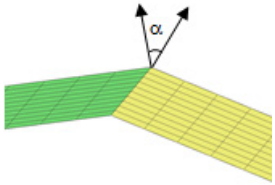
FOLD-LINE SETTINGS

A fold-line is an edge where the boundaries that meet do not have a continuous orientation of the normal vector. Along fold lines, the degrees of freedom are in general not continuous, but rather connected by a set of constraints.

Select a **Constraint type** — **Full** or **Simplified**. The full fold-line constraint formulation is accurate, consistent with the degrees of freedom, and does not produce any stress singularities near the fold-lines. However, it is computationally expensive, and leads to a nonlinear problem, even in an otherwise linear model. On the other hand, the simplified fold-line constraint formulation assumes that thin shell kinematics can be used. It is computationally more efficient, does not force the problem to be nonlinear, and works very well for thin or moderately thick shells.

The fold-line limit angle α is the smallest angle between the normals of two boundaries that makes their intersection to be treated as a fold line. The normal to the layered shell is discontinuous along a fold-line. Enter a value or expression in the α field. The default

value is 0.05 radians (approximately 3°). The value must be larger than 0, and less than $\pi/2$, but angles larger than a few degrees are not usually meaningful.



See Fold-Line Connection in the Structural Mechanics Theory chapter.

DEFAULT THROUGH-THICKNESS RESULT LOCATION

Enter a number between -1 and 1 for the **Local z-coordinate [-1,1] for thickness-dependent results** Z. The value can be changed from −1 (downside) to +1 (upside). The default is +1. A value of 0 means the midsurface of the layered shell. This is the default position for stress and strain evaluation during the results analysis.

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. Normally these settings do not need to be changed.

You can choose how extra ODE variables added by some features are grouped in the **Dependent Variables** node of a generated solver sequence.

- Select the **Rigid materials** checkbox to group variables added by Rigid Material nodes.
- Select the **Rigid connectors** checkbox to group variables added by Rigid Connector and Rigid Connector, Interface nodes.
- Select the **Attachments** checkbox to group variables added by Attachment nodes.

The selection made in the **Advanced Settings** section can be overridden by the settings in the **Advanced** section of the Rigid Material, Rigid Connector, Rigid Connector, Interface, or Attachment features.

DISCRETIZATION

In the Layered Shell interface you can choose not only the order of the discretization, but also the type of shape functions: *Lagrange* or *serendipity*. For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will however give significant reductions of the model size for a given mesh containing hexahedral, prism, or quadrilateral elements. The default is to use **Quadratic Lagrange** shape functions for the **Displacement field**.

The order of the discretization is used not only in the reference surface but also in the thickness direction. It is possible to choose different order of discretization in the reference surface and in the thickness direction.

See Discretization for the details on discretization order for a layered shell.

DEPENDENT VARIABLES

The physics interface uses the global spatial components of the **Displacement field** u as dependent variables. The default names for the components are (u, v, w).

Since the Layered Shell Interface uses a discretization also in the thickness direction, the degrees of freedom are allocated not only in the plane of the shell, but also in that direction.

You can change both the field name and the individual component names. If a new field name coincides with the name of another displacement field, the two fields (and the interfaces which define them) share degrees of freedom and dependent variable component names.

A new field name must not coincide with the name of a field of another type (that is, it must contain a displacement field), or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

- Domain, Boundary, Edge, Point, and Pair Nodes for the Layered Shell Interface
- Theory for the Layered Shell Interface

- *Micromechanics and Macromechanics of a Composite Cylinder*: Application Library path **Composite_Materials_Module/Tutorials/composite_cylinder_micromechanics_and_stress_analysis**
- *Forced Vibration Analysis of a Composite Laminate*: Application Library path **Composite_Materials_Module/Dynamics_and_Vibration/forced_vibration_of_a_composite_laminate**

The Shell Properties and Interface Selection Sections

The physics features can be of two, fundamentally different, types: Those who act on the layers themselves, and those who act on the interfaces between layers. The settings for these two types of features are slightly different.

THE SHELL PROPERTIES SECTION

Almost all layer type of physics nodes that you add in the Layered Shell interface have a section named **Shell Properties**. Selections in this section interact with the standard selections of geometrical objects (boundary, edges, or points) in order to provide a complete specification of where a material property or boundary condition is to be applied. The default selection is typically inherited from the selection in the physics interface, or from a parent node.

The **Use all layers** checkbox indicates that all layers within the geometrical selection are used as selection. Only layers that are selected in the physics interface and parent nodes are considered, though.

When **Use all layers** is deselected, you will be presented a list of all applicable layups defined by **Layered Material Link**, **Layered Material Stack**, and **Single Layer Material** nodes in the component. If the parent node has **Use all layers** selected, all such nodes are selectable. If not, the same selection as in the parent is the only option.

When a single layup is selected, you can usually select individual layers within it. There are some exceptions, because for some physics features it is not meaningful not to act on all layers.

THE INTERFACE SELECTION SECTION

Almost all interface type of physics nodes that you add in the Layered Shell interface have a section named **Interface Selection**. Selections in this section interact with the standard selections of geometrical objects (boundary, edges, or points) in order to

provide a complete specification of where a material property or boundary condition is to be applied.

You can only select interfaces that are adjacent to a layer selected in the parent. Some boundary conditions also impose other restrictions for physical reasons.

The interface selection in the **Apply to** list also provides a number of shortcuts:

- **Top interface**
- **Bottom interface**
- **Exterior interfaces**
- **Interior interfaces**
- **All interfaces**

If all layers in the laminate are selected by the parent, then **Exterior interfaces** means top and bottom. If not, it means all interfaces that are 'free' in the sense that they have a selected layer on one side only.

If all layers in the laminate are selected by the parent, then **Interior interfaces** means all interfaces except top and bottom. If not, it means all interfaces that have a selected layer on both sides.

When **All interfaces** is selected, it applies to all interfaces of layers selected in the parent.

See Physics Interface Selection for more details.

For a general description of layer selections, see Layer and Interface Selections in the documentation for the Composite Materials Module.

Domain, Boundary, Edge, Point, and Pair Nodes for the Layered Shell Interface

The Layered Shell Interface have the following domain, boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Layered Shell interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Boundary** level are

- **Material Models**
- **Face and Volume Loads**
- **Mass, Spring, and Damper**
- **Face and Domain Constraints**

The submenus at the **Edge** level are

- **Connections**
- **Mass, Spring, and Damper**
- **More Constraints**

There is also the **Points** submenu.

In order to distinguish the layer features with interface features, all the interface features are further grouped in an **Interfaces** menu and placed within the respective submenu.

LINKS TO FEATURE NODE INFORMATION

These nodes are described in this section (listed in alphabetical order).

These nodes are described in the documentation for the Solid Mechanics interface:

- Thermal Expansion (Rigid Connector)
- Applied Force (Rigid Connector)
- Applied Force (Rigid Material)
- Applied Moment (Rigid Connector)
- Applied Moment (Rigid Material)
- Base Excitation
- Contact
- Fixed Constraint (Rigid Material)
- Gravity
- Initial Values (Rigid Material)
- Mass and Moment of Inertia (Rigid Connector)
- Mass and Moment of Inertia (Rigid Material)
- Prescribed Displacement/Rotation
- Spring Foundation (Rigid Material)

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics Reference Manual*

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Initial Values

The **Initial Values** node adds initial values for the displacement and velocity fields. It can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. In addition to the default **Initial Values** node always present in the interface, you can add more **Initial Values** nodes if needed.

SHELL PROPERTIES

Select the material layers to which the initial values are to be applied.

TABLE 7-1: LAYER SELECTIONS; INITIAL VALUES.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added. In the default material node, the setting is always inherited from the interface setting and cannot be changed.	Not available

- All other settings are described in the documentation for Initial Values in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

Context Menus

Layered Shell > **Initial Values**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Layered Shell** > **Initial Values**

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic layered shell and an interface for defining the elastic material properties. When working with layered shells, you almost invariably take the material data from what has been defined using **Layered Material Link**, **Layered Material Stack**, or **Single Layer Material** nodes. It is however possible to override some data from a **Linear Elastic Material** node too.

In order to have a correct model, all layers must have been assigned material data for all boundaries selected in the settings for the interface. The override rules for the material models in the Layered Shell interface cannot enforce this, in the same way as

for other physics interfaces. You can have several **Linear Elastic Material** nodes with the same (or partially overlapping) geometrical selections, but with different layer selections.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Activation
- Safety

SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 7-2: LAYER SELECTIONS; LINEAR ELASTIC MATERIAL.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added. In the default material node, the setting is always inherited from the interface setting and cannot be changed.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the properties of the layers differ in the sense of the **Material symmetry** setting (**Isotropic**, **Orthotropic**, or **Anisotropic**) you can still use a single **Linear Elastic Material** node. Select the most complex of the models. Isotropic material data available in a **Material** node will automatically be converted to the two other types if needed. Similarly, orthotropic data will automatically be converted to anisotropic.

Data given in the other sections of this node applies to all selected layers. Thus, if you enter material data explicitly, rather relying on the default **From material** option, you will override that material property for all selected layers.

- All other settings are described in the documentation for Linear Elastic Material in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Material Models** > **Linear Elastic Material**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Material Models** > **Linear Elastic Material**

Hyperelastic Material

The **Hyperelastic Material** subnode adds the equations for hyperelasticity at large strains. Hyperelastic materials can be suitable for modeling rubber and other polymers, biological tissue, and also for applications in acoustoelasticity.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

When a hyperelastic material is included in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** checkbox in the study settings is selected and cannot be cleared.

By adding the following subnodes to the **Hyperelastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- External Stress
- External Strain
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Polymer Viscoplasticity
- Mullins Effect

SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 7-3: LAYER SELECTIONS; HYPERELASTIC MATERIAL.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Data given in the other sections of this node applies to all layers. Thus, if you enter material data explicitly, rather relying on the default **From material** option, you will override that material property for all selected layers.

- All other settings are described in the documentation for Hyperelastic Material in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Material Models** > **Hyperelastic Material**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Material Models** > **Hyperelastic Material**

Piezoelectric Material

The **Piezoelectric Material** node defines the piezoelectric material properties either in stress–charge form using the elasticity matrix and the coupling matrix, or in strain–charge form using the compliance matrix and the coupling matrix. It is normally used together with a **Piezoelectricity, Layered** multiphysics coupling node and a corresponding **Piezoelectric Layer** node in the **Electric Currents in Layered Shells** interface. This node is added by default to the **Layered Shell** interface when adding a **Piezoelectricity, Layered Shell** multiphysics interface.

The **Piezoelectric Material** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

When working with layered shells, you almost invariably take the material data from what has been defined using **Layered Material Link**, **Layered Material Stack**, or **Single Layer Material** nodes. It is however possible to override some data from a **Piezoelectric Material** node.

In order to have a correct model, all layers must have been assigned material data for all boundaries selected in the settings for the interface. The override rules for the material models in the Layered Shell interface cannot enforce this, in the same way as for other physics interfaces. You can have several **Piezoelectric Material** nodes with the same (or partially overlapping) geometrical selections, but with different layer selections.

By adding the following subnodes to the **Piezoelectric Material** node you can incorporate many other effects:

- Initial Stress and Strain
- Thermal Expansion (for Materials)
- Mechanical Damping
- Coupling Loss

- Dielectric Loss

When the **Piezoelectric Material** node is added to the Layered Shell interface in the absence of an active **Piezoelectricity, Layered** multiphysics coupling node, the material behaves similarly to a Linear Elastic Material node. The elastic properties will correspond to the elasticity or compliance matrix entered (see below). The piezoelectric effect is then *not* included in the equation system.

See also Piezoelectricity in the *Structural Mechanics Theory* chapter.

SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 7-4: LAYER SELECTIONS; PIEZOELECTRIC MATERIAL.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Data given in the other sections of this node applies to all layers. Thus, if you enter material data explicitly, rather relying on the default **From material** option, you will override that material property for all selected layers.

PIEZOELECTRIC MATERIAL PROPERTIES

Select a **Constitutive relation** — **Stress–charge form** or **Strain–charge form**. For each of the following, the default uses values **From material**. For **User defined** enter other values in the matrix or field as needed.

- For **Stress–charge form**, select an **Elasticity matrix, Voigt notation** (c_{E}).
- For a **Strain–charge form**, select a **Compliance matrix, Voigt notation** (s_{E}).
- Select a **Coupling matrix, Voigt notation** (d).
- Select a **Relative permittivity** (e_{rS} or e_{rT}).

- Enter values for the **Remanent electric displacement** D_r.
- Select a **Density** (p).

For entering these matrices, use the following order (Voigt notation), which is the common convention for piezoelectric materials: xx, yy, zz, yz, xz, zy.

Density

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

The density is for dynamic analysis, and also when computing mass forces for gravitational or rotating frame loads.

See also

- Mass Density and Volume Reference Temperature.
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

OUT-OF-PLANE MATERIAL ORIENTATION

The layered material always operates with a boundary coordinate system on the base surface (laminate system). For such systems, the third base vector direction is always normal to the surface. Use a special control available in this section if you need to change the out-of-plane orientation of the material. This is essential if your piezoelectric device requires the pole direction to be tangential to the shell, but the pole direction in the material data coincides with the third coordinate axis — such material orientation is assumed for all the piezoelectric material data available in COMSOL Material Library.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by **Mechanical damping**.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the integration points for the weak contribution of the feature. Select a method for **Hourglass stabilization** — **Automatic**, **Manual**, or **None** to use in combination with the reduced integration scheme. The default **Automatic** stabilization technique is based on the shape function and shape order of the displacement field.

Control the hourglass stabilization scheme by using the **Manual** option. Select **Shear stabilization** (default) or **Volumetric stabilization**.

When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}, and the shear correction factor k_{stb}. The value for G_{stb} should be in the order of magnitude of the equivalent shear modulus.

When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

- Modeling Piezoelectric Problems
- Modeling Geometric Nonlinearity

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Material Models** > **Piezoelectric Material**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Material Models** > **Piezoelectric Material**

Viscoelasticity

Use the **Viscoelasticity** subnode to add viscous stress contributions to an elastic material model.

SHELL PROPERTIES

Select the layer or layers for which this viscoelastic model is to be used.

TABLE 7-5: LAYER SELECTIONS; VISCOELASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the viscoelastic model differs between layers, you will need to add several **Viscoelasticity** nodes with different layer selections.

If the same layer is selected in two **Viscoelasticity** nodes being active on the boundary, then there will be two viscoelastic contributions.

- All other settings are described in the documentation for Viscoelasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Viscoelasticity**
Layered Shell > **Hyperelastic Material** > **Viscoelasticity**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Viscoelasticity**

Mullins Effect

Use the **Mullins effect** subnode to define the properties for modeling the stress-softening phenomenon under cyclic loading. The Mullins effect can be used together with Hyperelastic Material.

SHELL PROPERTIES

Select the layer or layers for which this Mullins effect model is to be used.

TABLE 7-6: LAYER SELECTIONS; MULLINS EFFECT.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the Mullins effect model differs between layers, you will need to add several **Mullins Effect** nodes with different layer selections. If there are two **Mullins Effect** nodes where the same layer is selected on the same boundary, the latter will be overridden on the common selections.

- All other settings are described in the documentation for Mullins Effect in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Hyperelastic Material** > **Mullins Effect**

Ribbon

Physics tab with **Hyperelastic Material** node selected in the model tree:

Attributes > **Mullins Effect**

Plasticity

Use the **Plasticity** subnode to define the properties for modeling elastoplastic materials. This material model can be used together with the Linear Elastic Material or Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which this plasticity model is to be used.

TABLE 7-7: LAYER SELECTIONS; PLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the material data values differ in every layer, you can use a single **Plasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If there are two **Plasticity** nodes where the same layer is selected on the same boundary, the latter will be overridden on the common selections

- All other settings are described in the documentation for Plasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

PLASTICITY MODEL

Use this section to define the plastic properties of the material.

Plasticity Model

The settings are the same as described in Plasticity Model for Solid Mechanics.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Plasticity**

Layered Shell > **Hyperelastic Material** > **Plasticity**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Plasticity**

Set Variables

Use the **Set Variables** subnode to Plasticity of a Linear Elastic Material or Hyperelastic Material to reset plasticity variables according to a **Setting condition** that you define. When the **Setting condition** is satisfied, the plasticity variables are reset to the specified values.

SHELL PROPERTIES

Select the layer or layers for which the variables are to be defined.

TABLE 7-8: LAYER SELECTIONS; SAFETY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

SET VARIABLES

Enter the **Setting condition**. This is a Boolean expression that will determine when the plastic variables are reset.

From the **Equivalent plastic strain** list, select **Do not set** or **User defined**. The default **User defined** value is zero.

From the **Plastic strain tensor** list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components of the **Plastic strain tensor**.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Plasticity** > **Set Variables**
Layered Shell > **Hyperelastic Material** > **Plasticity** > **Set Variables**

Ribbon

Physics tab with **Plasticity** node selected in the model tree:

Attributes > **Set Variables**

Creep

Use the **Creep** subnode to define the creep properties of a Linear Elastic Material or Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which this creep model is to be used.

TABLE 7-9: LAYER SELECTIONS; CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the creep model differs between layers, you will need to add several **Creep** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Creep** nodes being active on the boundary, then there will be two creep strain contributions.

- All other settings are described in the documentation for Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Creep**

Layered Shell > **Hyperelastic Material** > **Creep**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Creep**

Additional Creep

Use the **Additional Creep** subnode to define additional contributions to the creep model defined by the parent Creep node, such as primary or tertiary creep behavior. A **Creep** node can have any number of **Additional Creep** subnodes with different settings to model advanced creep mechanisms.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which the additional creep model is to be used.

TABLE 7-10: LAYER SELECTIONS; ADDITIONAL CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the creep model differs between layers, you will need to add several **Additional Creep** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Additional Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Creep** nodes being active on the boundary, then there will be two creep strain contributions

- All other settings are described in the documentation for Additional Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Creep** > **Additional Creep**

Layered Shell > **Hyperelastic Material** > **Creep** > **Additional Creep**

Ribbon

Physics tab with **Creep** node selected in the model tree:

Attributes > **Additional Creep**

Viscoplasticity

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which this viscoplasticity model is to be used.

TABLE 7-11: LAYER SELECTIONS; VISCOPLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the viscoplasticity model differs between layers, you will need to add several **Viscoplasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Viscoplasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Viscoplasticity** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Viscoplasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Viscoplasticity**
Layered Shell > **Hyperelastic Material** > **Viscoplasticity**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Viscoplasticity**

Polymer Viscoplasticity

Use the **Polymer Viscoplasticity** subnode to define the viscoplastic properties of a Hyperelastic Material node.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which this viscoplastic model is to be used.

TABLE 7-12: LAYER SELECTIONS; POLYMER VISCOPLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the viscoplastic model differs between layers, you will need to add several **Polymer Viscoplasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Polymer Viscoplasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Polymer Viscoplasticity** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Polymer Viscoplasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Hyperelastic Material** > **Polymer Viscoplasticity**

Ribbon

Physics tab with **Hyperelastic Material** node selected in the model tree:

Attributes > **Polymer Viscoplasticity**

Additional Network

Use the **Additional Network** subnode to define additional contributions to the Parallel Network model defined by the parent Polymer Viscoplasticity node. A **Polymer Viscoplasticity** node can have any number of **Additional Network** subnodes with different settings to model advanced viscoplastic mechanisms.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which this creep model is to be used.

TABLE 7-13: LAYER SELECTIONS; ADDITIONAL NETWORK.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the *Parallel Network* model differs between layers, you will need to add several **Additional Networks** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Additional Network** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Additional Network** nodes being active on the boundary, then there will be two additional contributions

- All other settings are described in the documentation for Additional Network in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Hyperelastic Material** > **Polymer Viscoplasticity** > **Additional Network**

Ribbon

Physics tab with **Polymer Viscoplasticity** node selected in the model tree:

Attributes > **Additional Network**

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

SHELL PROPERTIES

Select the layer or layers for which this thermal strain contribution is to be added.

TABLE 7-14: LAYER SELECTIONS; THERMAL EXPANSION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the **Input type** differs between layers, you will need to add several **Thermal Expansion** nodes with different layer selections. If only the material data values differ, you can use

a single **Thermal Expansion** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Thermal Expansion** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Thermal Expansion (for Materials) in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the expansion with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Thermal Expansion**
Layered Shell > **Hyperelastic Material** > **Thermal Expansion**
Layered Shell > **Piezoelectric Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material**, **Hyperelastic Material**, or **Piezoelectric Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_{\mathrm{h}} M_m (c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, $\mathrm{M_m}$ is the molar mass, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration.

You can provide a concentration with a through-thickness variation by explicitly or implicitly use expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

SHELL PROPERTIES

Select the layer or layers for which this swelling strain contribution is to be added.

TABLE 7-15: LAYER SELECTIONS; HYGROSCOPIC SWELLING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the swelling properties differ between layers, you will need to add several **Hygroscopic Swelling** nodes with different layer selections. If only the material data values differ, you can use a single **Hygroscopic Swelling** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Hygroscopic Swelling** nodes being active on the same boundary, the second definition will override the previous.

Hygroscopic strains are proportional to the concentration, while structural strains are computed from the gradient of the displacement field. It is good practice to match the discretization order of hygroscopic and structural strains.

When adding a **Hygroscopic Swelling** subnode, and the concentration field is computed by another physics interface; use a discretization one order lower for the concentration field than what is used for the displacement field.

- All other settings are described in the documentation for Hygroscopic Swelling in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the swelling with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Hygroscopic Swelling**
Layered Shell > **Hyperelastic Material** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to the Linear Elastic Material, in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense but rather a contribution to the constitutive relation.

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

For details about initial stresses and strains, see Initial Stress and Strain.

SHELL PROPERTIES

Select the layer or layers for which this stress or strain contribution is to be added.

TABLE 7-16: LAYER SELECTIONS; INITIAL STRESS AND STRAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the initial stress or strain differ between layers, you will need to add several **Initial Stress and Strain** nodes with different layer selections.

If the same layer is selected in two **Initial Stress and Strain** nodes being active on the same boundary, the contributions will be added.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Initial stresses and strains are specified along the axes of the selected coordinate system.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Initial Stress and Strain in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe stress or strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode to the Linear Elastic Material or Hyperelastic Material in order to specify an additional stress contribution which is not part of the

constitutive relation. The external stress can be added to the total stress tensor, act as an extra load contribution, or behave as a residual stress.

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor, whereas the option to use the stress as only a load contribution exists only in **External Stress**.

SHELL PROPERTIES

Select the layer or layers for which this stress contribution is to be added.

TABLE 7-17: LAYER SELECTIONS; EXTERNAL STRESS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. External stresses are specified along the axes of the selected coordinate system.

If the external stress input type differs between layers, you will need to add several **External Stress** nodes with different layer selections.

If the same layer is selected in two **External Stress** nodes being active on the same boundary, the contributions will be added.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for External Stress in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe stress values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **External Stress**
Layered Shell > **Hyperelastic Material** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model builder tree:

Attributes > **External Stress**

External Strain

The **External Strain** subnode allows you to provide inelastic strain contributions to the Linear Elastic Material or Hyperelastic Material on a variety of formats.

SHELL PROPERTIES

Select the layer or layers for which this strain contribution is to be added.

TABLE 7-18: LAYER SELECTIONS; EXTERNAL STRAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. External strains are specified along the axes of the selected coordinate system.

If the external strain input type differs between layers, you will need to add several **External Strain** nodes with different layer selections.

If the same layer is selected in two **External Strain** nodes being active on the same boundary, the contributions will be added.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for External Strain in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **External Strain**

Layered Shell > **Hyperelastic Material** > **External Strain**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model builder tree:

Attributes > **External Strain**

Inelastic Strain Rate

The **Inelastic Strain Rate** subnode allows you to provide inelastic strain contributions to the material models **Linear Elastic Material**, **Nonlinear Elastic Material**, and **Hyperelastic Material** on a variety of formats as in External Strain. Here, however, the inelastic contribution is given as a rate, and the total inelastic contribution is computed by integrating this rate in time.

SHELL PROPERTIES

Select the layer or layers for which this strain contribution is to be added.

TABLE 7-19: LAYER SELECTIONS; EXTERNAL STRAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Inelastic strains are specified along the axes of the selected coordinate system.

If the inelastic strain input type differs between layers, you will need to add several **Inelastic Strain Rate** nodes with different layer selections.

If the same layer is selected in two **Inelastic Strain Rate** nodes being active on the same boundary, the contributions will be added.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Inelastic Strain Rate in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Inelastic Strain Rate**
Layered Shell > **Hyperelastic Material** > **Inelastic Strain Rate**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model builder tree:

Attributes > **Inelastic Strain Rate**

Damage

The **Damage** subnode allows you to model damage and cracking in brittle materials according to various criteria. It is available in the Solid Mechanics, Shell, and Layered Shell interfaces, and it can be used in combination with the Linear Elastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

SHELL PROPERTIES

Select the layer or layers for which the damage is to be computed.

TABLE 7-20: LAYER SELECTIONS; DAMAGE.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the damage input type differs between layers, you will need to add several **Damage** nodes with different layer selections.

If the same layer is selected in two **Damage** nodes being active on the same boundary, the contributions will be overridden.

- All other settings are described in the documentation for Damage in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

See also

- Modeling Damage in the *Structural Mechanics Modeling* chapter.
- Damage Models in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Damage**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Damage**

Activation

Use the **Activation** subnode to activate or deactivate layers according to an **Activation expression** that you define. When the activation expression is satisfied, the material is activated in a state of zero stress.

SHELL PROPERTIES

Select the layer or layers for which this stress contribution is to be added.

TABLE 7-21: LAYER SELECTIONS; ACTIVATION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the same layer is selected in two **Activation** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Activation in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe activation properties with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Activation**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model builder tree:

Attributes > **External Strain**

Delamination

By adding a **Delamination** node, you can model interfacial failure between the layers in a laminate.

The formulation of the delamination models is essentially identical to the decohesion model used in contact simulations. See also:

- Including Adhesion and Decohesion, and The Decohesion Node in the *Structural Mechanics Modeling* chapter.
- Decohesion in the *Structural Mechanics Theory* chapter.

INTERFACE SELECTION

Select the interface or interfaces for which this delamination model is to be added.

TABLE 7-22: INTERFACE SELECTIONS; DELAMINATION.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Interior interfaces**. The choice **All interfaces** is equivalent to **Interior interfaces**.	When **Selected Interfaces** is selected. Only internal interfaces between layers selected in the physics interface can be selected.

If the same interface is selected in two **Delamination** nodes being active on the same boundary, the second definition will override the previous.

For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

COORDINATE SYSTEM SELECTION

The interface stresses are defined as boundary tractions with respect to the selected coordinate system. The selection is limited to boundary systems. Make sure that the tangents of the selected boundary system are well defined.

INITIAL STATE

Select an initial state for the interface, **Bonded** or **Delaminated**. If the state is delaminated, the effect of this node is only to act as a contact condition between the

two delaminated layers. In this case, only the **Contact** section of the settings is displayed.

ADHESION

In this section, you specify the stiffness of the adhesive layer in the normal and tangential directions. The adhesive normal stiffness is used in tension only; in compression the penalty factor is always used.

The default is to set **Adhesive stiffness** to **User defined**. For this setting you enter each component of the stiffness vector $\mathbf{k}_A$ individually.

When **Adhesive stiffness** is set to **Use material data**, the stiffness vector $\mathbf{k}_A$ is computed from material data and layer thickness. From the **Specify** list, select a pair of elastic properties — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, or **Bulk modulus and shear modulus**. For the chosen properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

Enter a **Thickness**, d_s, to specify the physical thickness of the adhesive layer.

DELAMINATION

Select a **Cohesive zone model** — **Displacement-based damage** or **Energy-based damage** to choose the type of variable that controls the damage process.

Select a **Traction separation law** — **Linear**, **Exponential**, **Polynomial**, or **Multilinear**. The definition of these differ between the two cohesive zone models, and the last option is available only for **Displacement-based damage**.

For the displacement-based damage models, enter:

- **Tensile strength**, σ_t. This is the peak stress in pure tension.
- **Shear strength**, σ_s. This is the peak stress in pure shear.
- **Tensile energy release rate**, G_{ct}. This is the energy released during the whole delamination process in a state of pure tension.
- **Shear energy release rate**, G_{cs}. This is the energy released during the whole delamination process in a state of pure shear.

For **Multilinear separation**, also enter the **Shape factor**, λ.

When the traction separation law is **Linear**, **Exponential**, or **Polynomial** select the **Mixed mode criterion** to be either **Power law** or **Benzeggagh-Kenane**. In either case, enter the

Mode mixity exponent α. The mixed mode criterion determines how normal and shear components are combined into a single scalar failure criterion. For the **Multilinear** separation law, the mixed mode criterion is always linear (equivalent to a power law with $\alpha = 1$.)

For the energy-based damage models, enter:

- **Tensile damage threshold**, G_{0t}. This is the elastic energy at the onset of damage in pure tension.
- **Shear damage threshold**, G_{0s}. This is the elastic energy at the onset of damage in pure shear.
- **Tensile energy release rate**, G_{ct}. This is the energy released during the whole delamination process in a state of pure tension.
- **Shear energy release rate**, G_{cs}. This is the energy released during the whole delamination process in a state of pure shear.
- **Mode mixity exponent, damage initiation**, α_0. The value determines how normal and shear components are combined into a single scalar criterion for damage initiation.
- **Mode mixity exponent**, α_c. The value determines how normal and shear components are combined into a single scalar failure criterion.
- **Smoothening parameter**, N. This parameter adjusts the shape of the traction separation law. It is only available for the **Exponential** and **Polynomial** options. By default, $N = 1$; a smaller value gives a smoother behavior.

In the **Regularization** list it is possible to add a viscous delay to the damage growth for time-dependent studies. Do this by selecting **Delayed damage** and enter a value for the **Characteristic time**, τ.

CONTACT

In this section you specify the **Penalty factor** used to minimize the overclosure of the two layers during compression in the normal direction. The default is to use **From adhesive stiffness**, in which case the normal component of the stiffness vector $\mathbf{k}_A$ is used.

When **Penalty factor** is set to **User defined**, enter the **Contact pressure penalty factor** p_n. The default value is `<phys>.Eequ/<phys>.d_ad`.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Maximum damage determines the residual stiffness of the adhesive layer after delamination. By default, $d_{max} = 1$, which means that no residual stiffness remains. Enter a value smaller than 1 to introduce some residual stiffness.

Select **Compute damage dissipation energy** to compute and store to the energy dissipated by damage.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Material Models** > **Delamination**

Ribbon

Physics tab with **Layered Shell** selected in the Model Builder tree:

Boundaries > **Delamination**

Safety

Add the **Safety** subnode to a Linear Elastic Material to set up variables which can be used to check the risk of failure according to various criteria. Four different variables describing the failure risk are defined, as described in Table 7-23

You can add any number of **Safety** nodes to a single material model. The contents of this feature do not affect the analysis results as such, so you can add **Safety** nodes after having performed an analysis, and just do an **Update Solution** in order to access to the new variables for result evaluation.

TABLE 7-23: VARIABLES FOR SAFETY FACTOR EVALUATION.

VARIABLE	DESCRIPTION	CRITERION FULFILLED	CRITERION VIOLATED
Failure index, FI	For a linear criterion, this is the ratio between the computed value and the given limit.	FI<1	FI>1
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=1
Safety factor, SF	For a linear criterion, this is 1/FI.	SF>1	SF<1
Margin of safety, MoS	SF-1	MoS>0	MoS<0

For orthotropic and anisotropic failure criteria, the directions are given by the coordinate system selection in the parent node.

SHELL PROPERTIES

Select the layer or layers for which the safety factor variables are to be defined.

TABLE 7-24: LAYER SELECTIONS; SAFETY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Each **Safety** subnode defines its own set of variables, so there is no interaction if you add several such nodes with the same selection.

- All other settings are described in the documentation for Safety in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe parameters for the allowable values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Variables** > **Safety**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model builder tree:

Attributes > **Variables** > **Safety**

Damping

Using the **Damping** subnode, you can add several types of damping to the material model. Damping can be used in Time Dependent, Eigenfrequency, and Frequency Domain studies; for other study types the settings in the **Damping** subnode are ignored.

SHELL PROPERTIES

Select the layer or layers for which this damping model is to be used.

TABLE 7-25: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Damping** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Damping** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Damping in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe damping values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Linear Elastic Material** > **Damping**
Layered Shell > **Hyperelastic Material** > **Damping**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Damping**

Mechanical Damping

The **Mechanical Damping** subnode allows you to model mechanical losses in the Piezoelectric Material, either using loss factor material data for the stiffness, or in the form of Rayleigh proportional damping.

SHELL PROPERTIES

Select the layer or layers for which this damping model is to be used.

TABLE 7-26: LAYER SELECTIONS; MECHANICAL DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Mechanical Damping** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Mechanical Damping** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Mechanical Damping in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe damping values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Piezoelectric Material** > **Mechanical Damping**

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:

Attributes > **Mechanical Damping**

Coupling Loss

The **Coupling Loss** subnode allows you to model losses in the piezoelectric coupling in a Piezoelectric Material, either by using the loss factor material data for the coupling matrix or as Rayleigh proportional damping.

SHELL PROPERTIES

Select the layer or layers for which this loss model is to be used.

TABLE 7-27: LAYER SELECTIONS; COUPLING LOSS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the loss model differs between layers, you will need to add several **Coupling Loss** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Coupling Loss** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Coupling Loss in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe loss values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Piezoelectric Material** > **Coupling Loss**

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:

Attributes > **Coupling Loss**

Dielectric Loss

The **Dielectric Loss** subnode allows you to model losses in the electrical losses in a Piezoelectric Material. The losses can be prescribed either by using a loss factor for the electric permittivity, or in the form of dielectric dispersion.

SHELL PROPERTIES

Select the layer or layers for which this loss model is to be used.

TABLE 7-28: LAYER SELECTIONS; DIELECTRIC LOSS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the loss model differs between layers, you will need to add several **Dielectric Loss** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Dielectric Loss** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings are described in the documentation for Dielectric Loss in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe loss values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Piezoelectric Material** > **Dielectric Loss**

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:

Attributes > **Dielectric Loss**

Rigid Material

Add the **Rigid Material** node and select one or more boundaries to make them a rigid body. **Rigid Material** is a material model, with only one material property: the mass density. All layers through the shell thickness act together as a single rigid body, so it is not possible to select individual layers.

By default, an **Initial Values** node is added (see Initial Values (Rigid Material)).

You can add functionality to the rigid domain through the following subnodes:

- Fixed Constraint (Rigid Material) to fully constrain the rigid domain.
- Prescribed Displacement/Rotation to prescribe the displacement of individual degrees of freedom.
- Applied Force (Rigid Material) to apply a force in given point.
- Applied Moment (Rigid Material) to apply a moment.
- Mass and Moment of Inertia (Rigid Material) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Material) to add a translational or rotational spring or damper in a given point.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection.

TABLE 7-29: LAYER SELECTIONS; RIGID DOMAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	No

- All other settings are described in the documentation for Rigid Material in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Material Models** > **Rigid Material**

Ribbon
Physics tab with **Layered Shell** selected:

Boundaries > **Material Models** > **Rigid Material**

Free

The **Free** node is the default boundary condition. It means that there are no constraints and no loads acting on the edge. When the physics interface is added, a default **Free** node is added. If you look at the selections for this node, it will show all edges which do not have boundary conditions applied to all layers.

You can manually add **Free** nodes to override other boundary conditions. This is however seldom needed.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection.

TABLE 7-30: LAYER SELECTIONS; FREE.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

If you manually add a **Free** node, it will override loads and constraints on all layers from the selected edges.

For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus
Layered Shell > **Free**

Ribbon
Physics tab with a physics interface selected:

Edges > **Layered Shell** > **Free**

Prescribed Displacement

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in one or more directions to the selected geometric entity (boundary, edge, or point). It is also possible set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

If a displacement is prescribed in one direction, this leaves the material free to deform in the other directions.

To prescribe displacements on interfaces between the material layers, use Prescribed Displacement, Interface.

SHELL PROPERTIES

Select the layer or layers on which this constraint is to be used.

TABLE 7-31: LAYER SELECTIONS; PRESCRIBED DISPLACEMENT.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge, Point	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the physics interface can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Prescribed displacements are specified along the axes of the selected coordinate system.

If the same layer is selected in two **Prescribed Displacement** nodes being active on the same boundary, the second definition will override the previous.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Prescribed Displacement in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can prescribe displacements with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Prescribed Displacement**
Layered Shell > **Prescribed Displacement** (Edge)
Layered Shell > **Points** > **Prescribed Displacement**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Prescribed Displacement**
Edges > **Layered Shell** > **Prescribed Displacement**
Points > **Layered Shell** > **Prescribed Displacement**

Prescribed Displacement, Interface

The **Prescribed Displacement, Interface** node adds a condition where the displacements are prescribed in one or more directions to interfaces between material layers in the selected geometric entity (boundary, edge, or point). It is also possible set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

If a displacement is prescribed in one direction, this leaves the material free to deform in the other directions.

To prescribe displacements to the actual material layers, use Prescribed Displacement.

INTERFACE SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 7-32: INTERFACE SELECTIONS; PRESCRIBED DISPLACEMENT, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.
Edge, Point	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

If the same layer is selected in two **Prescribed Displacement, Interface** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Prescribed Displacement in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Displacement, Interface**

Layered Shell > **More Constraints** > **Interfaces** > **Prescribed Displacement, Interface** (Edge)

Layered Shell > **Points** > **Interfaces** > **Prescribed Displacement, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Prescribed Displacement, Interface**

Edges > **Interfaces** > **Prescribed Displacement, Interface**

Points > **Interfaces** > **Prescribed Displacement, Interface**

Prescribed Velocity

The **Prescribed Velocity** node adds a boundary or domain condition where the velocity is prescribed in one or more directions. This condition is applicable for time-

dependent and frequency domain studies. It is possible to prescribe a velocity in one direction, leaving the structure free in the other directions. The **Prescribed Velocity** node is a constraint and overrides any other constraint on the same selection.

To prescribe velocities on the top or bottom surfaces of the shell, use Prescribed Velocity, Interface.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection. The condition is applied all through the thickness of the shell.

TABLE 7-33: LAYER SELECTIONS; PRESCRIBED VELOCITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge	Same as interface selection when the node is added.	No

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Prescribed velocities are specified along the axes of the selected coordinate system.

If the same layer is selected in two **Prescribed Velocity** nodes being active on the same boundary, the second definition will override the previous.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Prescribed Velocity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can prescribe velocities with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Prescribed Velocity**

Layered Shell > **More Constraints** > **Prescribed Velocity** (Edge)

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Prescribed Velocity**

Edges > **More Constraints** > **Prescribed Velocity**

Prescribed Velocity, Interface

The **Prescribed Velocity, Interface** node adds a boundary condition on the top or bottom face of the shell where the velocity is prescribed in one or more directions. This condition is applicable for time- dependent and frequency domain studies. It is possible to prescribe a velocity in one direction, leaving the structure free in the other directions. The **Prescribed Velocity, Interface** node is a constraint and overrides any other constraint on the same selection.

To prescribe velocities on the actual material layers, use Prescribed Velocity.

INTERFACE SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 7-34: INTERFACE SELECTIONS; PRESCRIBED VELOCITY, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.

If the same layer is selected in two **Prescribed Velocity, Interface** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Prescribed Velocity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Velocity, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Prescribed Velocity, Interface**

Prescribed Acceleration

The **Prescribed Acceleration** node adds a boundary or domain condition where the acceleration is prescribed in one or more directions. This condition is applicable for time- dependent and frequency domain studies. It is possible to prescribe an acceleration in one direction, leaving the structure free in the other directions. The **Prescribed Acceleration** node is a constraint and overrides any other constraint on the same selection.

To prescribe accelerations on the top or bottom surfaces of the shell, use Prescribed Acceleration, Interface.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection. The condition is applied all through the thickness of the shell.

TABLE 7-35: LAYER SELECTIONS; PRESCRIBED ACCELERATION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge	Same as interface selection when the node is added.	No

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Prescribed accelerations are specified along the axes of the selected coordinate system.

If the same layer is selected in two **Prescribed Acceleration** nodes being active on the same boundary, the second definition will override the previous.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Prescribed Acceleration in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can prescribe accelerations with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Prescribed Acceleration**
Layered Shell > **More Constraints** > **Prescribed Acceleration** (Edge)

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Prescribed Acceleration**
Edges > **More Constraints** > **Prescribed Acceleration**

Prescribed Acceleration, Interface

The **Prescribed Acceleration, Interface** node adds a boundary condition on the top or bottom face of the shell where the acceleration is prescribed in one or more directions. This condition is applicable for time- dependent and frequency domain studies. It is possible to prescribe an acceleration in one direction, leaving the structure free in the other directions. The **Prescribed Acceleration, Interface** node is a constraint and overrides any other constraint on the same selection.

To prescribe accelerations on the actual material layers, use Prescribed Acceleration.

INTERFACE SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 7-36: INTERFACE SELECTIONS; PRESCRIBED ACCELERATION, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.

If the same layer is selected in two **Prescribed Acceleration, Interface** nodes being active on the same boundary, the second definition will override the previous.

- All other settings are described in the documentation for Prescribed Acceleration in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Prescribed Acceleration, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Prescribed Acceleration, Interface**

Fixed Constraint

Use the **Fixed Constraint** node to set all displacements on the selected geometric entity (boundary, edge, or point) to zero.

To constrain interfaces between the material layers, use Fixed Constraint, Interface.

SHELL PROPERTIES

Select the layer or layers on which this constraint is to be used.

TABLE 7-37: LAYER SELECTIONS; FIXED CONSTRAINT.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge, Point	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the physics interface can be selected.

- All other settings are described in the documentation for Fixed Constraint in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Fixed Constraint**
Layered Shell > **Fixed Constraint** (Edge)
Layered Shell > **Points** > **Fixed Constraint**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Fixed Constraint**
Edges > **Layered Shell** > **Fixed Constraint**
Points > **Layered Shell** > **Fixed Constraint**

Fixed Constraint, Interface

Use the **Fixed Constraint, Interface** to add a condition where the displacements constrained in all directions at interfaces between material layers in the selected geometric entity.

To constrain the actual material layers, use Fixed Constraint.

INTERFACE SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 7-38: INTERFACE SELECTIONS; PRESCRIBED DISPLACEMENT, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.
Edge, Point	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

- All other settings are described in the documentation for Fixed Constraint in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Fixed Constraint, Interface**
Layered Shell > **More Constraints** > **Interfaces** > **Fixed Constraint, Interface** (Edge)
Layered Shell > **Points** > **Interfaces** > **Fixed Constraint, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Fixed Constraint, Interface**
Edges > **Interfaces** > **Fixed Constraint, Interface**
Points > **Interfaces** > **Fixed Constraint, Interface**

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Fixed Constraint** or **Prescribed Displacement**, or to **Fixed Constraint, Interface** or **Prescribed Displacement, Interface**) to prescribe a deformation of the constrained boundary caused by changes in temperature of the surroundings. This makes it possible to reduce the stresses caused by such boundary conditions.

SHELL PROPERTIES

This section is only present when the parent node has layer selection.

TABLE 7-39: LAYER SELECTIONS; THERMAL EXPANSION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Same level as parent	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

INTERFACE SELECTION

This section is only present when the parent node has interface selection.

Select the interface or interfaces for which this thermal expansion is to be added.

TABLE 7-40: INTERFACE SELECTIONS; THERMAL EXPANSION.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Same level as parent	Same as parent selection when the node is added.	When **Selected Interfaces** is selected. Only a subset of the interfaces selected in the parent can be selected.

- All other settings are described in the documentation for Thermal Expansion (for Constraints) in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Fixed Constraint** > **Thermal Expansion**

Layered Shell > **Fixed Constraint, Interface** > **Thermal Expansion**

Layered Shell > **Prescribed Displacement** > **Thermal Expansion**

Layered Shell > **Prescribed Displacement, Interface** > **Thermal Expansion**

Ribbon

Physics tab with **Fixed Constraint**, **Fixed Constraint, Interface**, **Prescribed Displacement**, or **Prescribed Displacement, Interface** node selected in the model tree:

Roller

Use the **Roller** node to add a roller constraint as boundary condition on the boundaries formed by the edge of the shell. The displacement is zero in the direction perpendicular (normal) to the boundary, but the boundary is free to move in the tangential direction. A **Roller** condition is similar to a Symmetry condition, but the latter has more options.

To apply a roller condition on one of the faces of the shell, use Roller, Interface.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection. The constraint is applied all through the thickness of the shell.

TABLE 7-41: LAYER SELECTIONS; ROLLER.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

- All other settings are described in the documentation for Roller in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Roller**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Layered Shell** > **Roller**

Roller, Interface

Use the **Roller, Interface** node to add a roller constraint as boundary condition to one of the faces of the shell. The displacement is zero in the direction perpendicular (normal) to the boundary, but the boundary is free to move in the tangential direction.

To apply a roller conditions on the boundaries formed by the edge of the shell, use Roller.

SHELL PROPERTIES

Select the interface or interfaces at which the constraint is to be applied.

TABLE 7-42: INTERFACE SELECTIONS; PRESCRIBED ACCELERATION, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.

- All other settings are described in the documentation for Roller in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Domain Constraints** > **Interfaces** > **Roller, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Roller, Interface**

Symmetry

Use the **Symmetry** node to add a boundary condition that represents symmetry in the geometry and in the loads. A symmetry condition is free in the plane and fixed in the out-of-plane direction.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection. The constraint is applied all through the thickness of the shell.

TABLE 7-43: LAYER SELECTIONS; SYMMETRY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

- All other settings are described in the documentation for Symmetry in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Symmetry**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **More Constraints** > **Symmetry**

Antisymmetry

Use the **Antisymmetry** node adds a boundary condition for an antisymmetry boundary. An antisymmetry condition is fixed in the plane and free in the out-of-plane direction.

When applied to an edge the antisymmetry plane is formed by the normal to the boundary and the edge tangent.

SHELL PROPERTIES

The selection in this section only acts as a filter for the boundary selection. The constraint is applied all through the thickness of the shell.

TABLE 7-44: LAYER SELECTIONS; ANTISYMMETRY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

- All other settings are described in the documentation for Antisymmetry in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Antisymmetry**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **More Constraints** > **Antisymmetry**

Rigid Motion Suppression

The **Rigid Motion Suppression** node adds a minimum number of constraints required to suppress rigid body modes. The constraints are selected so that there will be no reaction forces if the external loads are self-equilibrating. If you need to constrain several objects that are not physically connected, you must add as many **Rigid Motion Suppression** nodes as there are disconnected objects.

SHELL PROPERTIES

As the constraints are added automatically to the geometrical selection, it is not possible to select individual layers. The selection in this section only acts as a filter for the boundary selection.

TABLE 7-45: LAYER SELECTIONS; RIGID MOTION SUPPRESSION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	No

All other settings are described in the documentation for Rigid Motion Suppression in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face Constraints** > **Rigid Motion Suppression**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Domain Constraints** > **Rigid Motion Suppression**

Body Load

Add a **Body Load** to boundaries for modeling general volumetric loads.

For loads caused by gravity or rotation, it is more convenient to use the Gravity, Linearly Accelerated Frame, and Rotating Frame nodes.

SHELL PROPERTIES

Select the layer or layers to which this load is to be applied.

TABLE 7-46: LAYER SELECTIONS; BODY LOAD.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the interface can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Body loads are specified along the axes of the selected coordinate system.

Contributions from several **Body Load** nodes having a common selection will be added to each other.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Body Load in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can prescribe loads with through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Volume Loads** > **Body Load**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Volume Loads** > **Body Load**

Face Load

Add a **Face Load** to boundaries to use it as a pressure or tangential force acting on a surface. The loads are defined in the given coordinate system.

INTERFACE SELECTION

Select the interface or interfaces at which the load is to be applied.

TABLE 7-47: INTERFACE SELECTIONS; FACE LOAD.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.

- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Face loads are specified along the axes of the selected coordinate system.

FORCE

Select a **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, or **Pressure**. Then enter values or expressions for the components in the matrix based on the selection and the space dimension.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.
- For **Total force**, COMSOL Multiphysics then divides the total force by the area of the boundaries where the load is active. Then it is applied in the same way as for a **Force per reference area**. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.
- For **Pressure**, a scalar input is given, and the orientation of the load is along the direction of the shell boundary normal. The sign convention is such that for a positive value of the pressure, the load acts toward the shell, when considered as a

solid object. Thus, the orientation of the load is flipped if it is moved from the top side to the bottom side.

TABLE 7-48: LOAD TYPES.

LOAD TYPE	VARIABLE	SI UNIT
Force per reference area	$\mathbf{f}_A$	N/m^2
Force per deformed area	$\mathbf{f}_a$	N/m^2
Total force	$\mathbf{F}_{tot}$	N
Pressure	p	Pa

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- The pressure load is a 'follower load'. The direction changes with deformation in a geometrically nonlinear analysis.
- After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface, it is also possible to choose a predefined load from this list.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Volume Loads** > **Face Load**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Volume Loads** > **Face Load**

Rotating Frame

Centrifugal, Coriolis, and Euler forces are "fictitious" volume forces that need to be introduced in a rotating frame of reference, since it is not an inertial system. Use a

Rotating Frame node to add the effect of these forces. The forces are generated by all selected features in the physics interface having a density, mass, or mass distribution. You select boundaries, and all edges and points which belong to the selection are automatically included.

SHELL PROPERTIES

As a frame rotation always acts on the entire structure, it is not possible to select individual layers. The selection in this section only acts as a filter for the boundary selection.

TABLE 7-49: LAYER SELECTIONS; ROTATING FRAME.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	No

- All other settings are described in the documentation for Rotating Frame in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Volume Loads** > **Rotating Frame**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Volume Loads** > **Rotating Frame**

Linearly Accelerated Frame

When you add a **Linearly Accelerated Frame** node, inertial forces are applied to all selected features in the physics interface with a density, mass, or mass distribution. You select boundaries, and all edges and points which belong to the selection are automatically included.

SHELL PROPERTIES

Select the layer or layers to which the frame acceleration force is to be applied.

TABLE 7-50: LAYER SELECTIONS; LINEARLY ACCELERATED FRAME.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the interface can be selected.

- All other settings are described in the documentation for Linearly Accelerated Frame in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Face and Volume Loads** > **Linearly Accelerated Frame**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Face and Volume Loads** > **Linearly Accelerated Frame**

Boundary Load

Use a **Boundary Load** to apply forces distributed on the layers along an edge of the layered shell. The load is defined in the given local coordinate system.

SHELL PROPERTIES

Select the layer or layers to which this load is to be applied.

TABLE 7-51: LAYER SELECTIONS; BOUNDARY LOAD.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the interface can be selected.

Contributions from several **Boundary Load** nodes having a common selection will be added to each other

- All other settings are described in the documentation for Boundary Load in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can prescribe loads with through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Boundary Load**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Layered Shell** > **Boundary Load**

Edge Load

Add an **Edge Load** as a force distributed on the interfaces between the layers along an edge of the layered shell. The load is defined in the given local coordinate system.

INTERFACE SELECTION

Select the interface or interfaces at which the load is to be applied.

TABLE 7-52: INTERFACE SELECTIONS; EDGE LOAD.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Edge	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

Contributions from several **Edge Load** nodes having a common selection will be added to each other

- All other settings are described in the documentation for Edge Load in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Edge Load**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Layered Shell** > **Edge Load**

Line Load

Use a **Line Load** to apply forces distributed along one or more lines in the through-thickness direction. In this case, a point selection represents the line or lines. The load is defined in the given local coordinate system.

SHELL PROPERTIES

Select the layer or layers to which this load is to be applied.

TABLE 7-53: LAYER SELECTIONS; LINE LOAD.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Point	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the interface can be selected.

Contributions from several **Line Load** nodes having a common selection will be added to each other

- All other settings are described in the documentation for Edge Load in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Points** > **Line Load**

Ribbon

Physics tab with **Layered Shell** selected:

Points > **Layered Shell** > **Line Load**

Point Load

Add an **Point Load** as a force on the interfaces between the layers on a point of the layered shell. The load is defined in the given local coordinate system.

INTERFACE SELECTION

Select the interface or interfaces at which the load is to be applied.

TABLE 7-54: INTERFACE SELECTIONS; POINT LOAD.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Point	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

Contributions from several **Point Load** nodes having a common selection will be added to each other

- All other settings are described in the documentation for Point Load in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Points** > **Interfaces** > **Point Load**

Ribbon

Physics tab with **Layered Shell** selected:

Points > **Interfaces** > **Point Load**

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response, the physics interface splits the harmonic load into two parameters:

- The amplitude, F, which is specified in the node for the load.
- The phase, ϕ, which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f:

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi f t + \phi)$$

SHELL PROPERTIES

Select the layer or interfaces based on this node is added under which type of load feature. You can select individual layers or interfaces in this node which are allowed on the parent node.

All other settings are described in the documentation for Phase in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Body Load** > **Phase**
Layered Shell > **Face Load** > **Phase**
Layered Shell > **Boundary Load** > **Phase**
Layered Shell > **Edge Load** > **Phase**
Layered Shell > **Line Load** > **Phase**
Layered Shell > **Point Load** > **Phase**

Ribbon

Physics tab with **Body Load**, **Face Load**, **Boundary Load**, **Edge Load**, **Line Load**, or **Point Load** node selected in the model tree:

Attributes > **Phase**

Spring Foundation

Use the **Spring Foundation** node to apply elastic and damping boundary conditions on the material layers (on boundaries, edges, and points) of a layered shell. For applying spring conditions on an interface, see Spring Foundation, Interface.

- When the selection is a boundary, a volumetric spring is added to the selected layers
- When the selection is an edge, a spring per area is added to the faces of the layers at the edge.
- When the selection is a point, a spring is added to the layer edges in the through-thickness direction for the selected layers.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

SHELL PROPERTIES

Select the material layers to which the spring condition is to be applied.

TABLE 7-55: LAYER SELECTIONS; SPRING FOUNDATION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge, Point	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the physics interface can be selected.

- All other settings are described in the documentation for Spring Foundation in the Solid Mechanics interface.
- For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter.
- For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Spring foundations are specified along the axes of the selected coordinate system.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Spring Foundation in the Solid Mechanics interface.
- For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter.
- For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Spring Foundation** (Boundary, Edge)
Layered Shell > **Points** > **Spring Foundation**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Spring Foundation**
Edges > **Mass, Spring, and Damper** > **Spring Foundation**
Points > **Layered Shell** > **Spring Foundation**

Spring Foundation, Interface

Use the **Spring Foundation, Interface** node to apply elastic and damping boundary conditions on the interfaces of material layers (on boundaries, edges, and points) of a layered shell. For applying spring conditions on a layer, see Spring Foundation.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

INTERFACE SELECTION

Select the interface or interfaces for which this spring condition is to be added.

TABLE 7-56: INTERFACE SELECTIONS; SPRING FOUNDATION, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.
Edge, Point	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

- All other settings are described in the documentation for Spring Foundation in the Solid Mechanics interface.
- For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter.
- For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Interfaces** > **Spring Foundation, Interface** (Boundary, Edge)

Layered Shell > **Points** > **Interfaces** > **Spring Foundation, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Spring Foundation, Interface**
Edges > **Interfaces** > **Spring Foundation, Interface**
Points > **Interfaces** > **Spring Foundation, Interface**

Thin Elastic Layer

Use the **Thin Elastic Layer** node to apply elastic and damping boundary conditions on the interior edges of a layered shell. Physically, this means that the spring condition is added at the interface between two layers located in adjacent boundaries.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

To apply elastic and damping boundary conditions on the interior boundaries of a layered shell, instead use the Thin Elastic Layer, Interface node.

The **Thin Elastic Layer** and Spring Foundation nodes are similar, with the difference that a Spring Foundation connects the structural part on which it is acting to a fixed "ground".

SHELL PROPERTIES

Select the material layers to which the spring condition is to be applied.

TABLE 7-57: LAYER SELECTIONS; THIN ELASTIC LAYER.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the physics interface can be selected.

COORDINATE SYSTEM SELECTION

The spring and damping constants are given with respect to the selected coordinate system, with the exception that when the stiffness is specified by **Use material data**, the coordinate system selection is not used.

SPRING

Select the **Spring type** and its associated spring constant or force using Table 7-58 as a guide. The default option is the spring type for the type of geometric entity and space dimension, and there are different combinations available based on this.

When the option is of the type “force as function of extension”, then the built-in variables describing the spring extension must be used in the expression, as described in Springs and Dampers. The spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix. In the Thin Elastic Layer, Interface node, **Use material data** is additionally available as **Spring type**.

For **Use material data**, the spring stiffness values are computed from material data and layer thickness. From the **Specify** list, select a pair of elastic properties — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, or **Bulk modulus and shear modulus**. Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another according to Table 4-8 in the *Solid Mechanics* chapter. For the chosen properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

Enter a **Thickness**, d_s, to specify the physical thickness of the elastic layer.

TABLE 7-58: SPRING TYPES FOR THE THIN ELASTIC LAYER FEATURE.

SPRING TYPE	VARIABLE	SI UNITS
Total spring constant	$\mathbf{k}_{tot}$	N/m
Spring constant per unit area	$\mathbf{k}_A$	$N/(m \cdot m^2)$
Total force as function of extension	$\mathbf{F}_{tot}$	N
Force per area as function of extension	$\mathbf{F}_A$	N/m^2
Use material data	N/A	N/A

ADDED MASS

Enter the added mass of the thin elastic layer. The type of input depends on the selected **Spring type**; see Table 7-58.

- For **Total spring constant** and **Total force as function of extension**, enter the **Total mass** m of the thin elastic layer.
- For **Spring constant per unit area** and **Force per area as function of extension**, enter the **Mass per unit area** ρ_A of the thin elastic layer.
- For **Use material data**, enter the **Density** ρ of the thin elastic layer. Either use the value **From material** or enter a **User defined** value or expression. In order to use **From material**, you must have assigned a material to the selected boundaries.

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s, which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS DAMPING

Select the **Damping type** using Table 7-59 as a guide. The default option is the default damping type for the space dimension. The damping matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **Full**. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 7-59: DAMPING TYPES FOR THE THIN ELASTIC LAYER FEATURE.

DAMPING TYPE	VARIABLE	SI UNITS
Damping constant per unit area	$\mathbf{d}_A$	N·s/(m·m^2)
Total damping constant	$\mathbf{d}_{tot}$	N·s/m

ENERGY DISSIPATION

You can select to compute and store energy dissipation variables in a time-dependent analysis. Doing so will add extra degrees of freedom to the model.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by damping.

- Dissipated Energy
- Energy Variables

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

	You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the *Structural Mechanics Modeling* chapter.

	• For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter. • For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter. • For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections. • You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Thin Elastic Layer** (Edge)

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Mass, Spring, and Damper** > **Thin Elastic Layer**

Thin Elastic Layer, Interface

Use the **thin Elastic Layer, Interface** node to apply elastic and damping boundary conditions on the interior boundaries of a layered shell.

By adding the Predeformation subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

INTERFACE SELECTION

Select the interface or interfaces for which this spring condition is to be added.

TABLE 7-60: INTERFACE SELECTIONS; THIN ELASTIC LAYER, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Interior interfaces**.	When **Selected Interfaces** is selected. Only interior interfaces can be selected.

- All other settings are described in the documentation for Thin Elastic Layer.
- For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter.
- For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Interfaces** > **Thin Elastic Layer** (Boundary)

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Interfaces** > **Thin Elastic Layer, Interface**

Predeformation

Use the **Predeformation** subnode to specify that the elastic forces in Spring Foundation, Spring Foundation, Interface, Thin Elastic Layer, or Thin Elastic Layer, Interface are nonzero at zero displacement. Thus, you can model cases where the unstressed state of the spring is in another configuration than the one described by the geometry.

The value of the predeformation can vary during the simulation, so it should not be interpreted as an initial value.

SHELL PROPERTIES

This section is only present when **Predeformation** is used as a subnode to **Spring Foundation** or **Thin Elastic Layer**.

Select the layers for which this predeformation value is to be used. You can only select a subset of the layers selected in the parent node.

INTERFACE SELECTION

This section is only present when **Predeformation** is used as a subnode to **Spring Foundation, Interface** or **Thin Elastic Layer, Interface**.

Select the interfaces for which this predeformation value is to be used. You can only select a subset of the interfaces selected in the parent node.

- All other settings are described in the documentation for Predeformation in the Solid Mechanics interface.
- For more information about the use of springs, see Springs and Dampers in the Structural Mechanics Modeling chapter.
- For a theoretical background, see Spring Foundation and Thin Elastic Layer in the Structural Mechanics Theory chapter.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Spring Foundation** > **Predeformation**
Layered Shell > **Spring Foundation, Interface** > **Predeformation**
Layered Shell > **Shin Elastic Layer** > **Predeformation**
Layered Shell > **Thin Elastic Layer, Interface** > **Predeformation**

Ribbon

Physics tab with **Spring Foundation**, **Spring Foundation, Interface**, **Thin Elastic Layer**, or **Thin Elastic Layer, Interface** node selected in the model tree:

Attributes > **Predeformation**

Added Mass

The **Added Mass** node is available for boundaries, edges, and points and can be used to supply inertia that is not part of the material itself. Such inertia does not need to be isotropic, in the sense that the inertial effects are not the same in all directions. The extra inertia is as a volumetric, boundary, or edge contribution to the selected material layers.

You can also add inertia to the interfaces between the material layers by using Added Mass, Interface.

SHELL PROPERTIES

Select the material layers to which this added mass contribution is to be applied.

TABLE 7-61: LAYER SELECTIONS; ADDED MASS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge, Point	Same as interface selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the physics interface can be selected.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains all applicable coordinate systems in the model, including the **Boundary system** and the **Layer local system**. The **Global coordinate system** is selected by default. Added masses are specified along the axes of the selected coordinate system.

- See Defining a Layer Local System in the *Composite Materials Modeling* chapter.
- All other settings are described in the documentation for Added Mass in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe added mass densities with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Added Mass** (Boundary, Edge)
Layered Shell > **Points** > **Added Mass** (Point)

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Mass, Spring, and Damper** > **Added Mass**
Edges > **Mass, Spring, and Damper** > **Added Mass**
Points > **Layered Shell** > **Added Mass**

Added Mass, Interface

The **Added Mass, Interface** node is available for boundaries and edges and can be used to supply inertia that is not part of the material itself. Such inertia does not need to be isotropic, in the sense that the inertial effects are not the same in all directions. The inertia is added at the selected interfaces between the material layers.

INTERFACE SELECTION

Select the interface or interfaces for which this added mass contribution is to be added.

TABLE 7-62: INTERFACE SELECTIONS; ADDED MASS, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary, Edge, Point	When added: **Top interface**.	When **Selected Interfaces** is selected. Any interface can be selected.

- All other settings are described in the documentation for Added Mass in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe added mass densities with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Mass, Spring, and Damper** > **Interfaces** > **Added Mass, Interface** (Boundary, Edge)

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Added Mass, Interface**

Edges > **Interfaces** > **Added Mass, Interface**

Adiabatic Heating

The **Adiabatic Heating** node adds the equations for adiabatic heating in layered shells caused by abrupt changes in temperature due to fast deformation.

The increase in temperature is then defined by solving the distributed equation

$$\rho C_p \dot{T} = \beta_{ah} Q_h$$

where ρ is the density, C_p is the heat capacity at constant pressure, T is the temperature field, β_{ah} is the coefficient of adiabatic heating, and Q_h corresponds to the heat sources due to mechanical dissipative processes.

Adiabatic heating is only available with some COMSOL products (see www.comsol.com/products/specifications/).

If the Composite Materials Module is available, adiabatic heating can be applied to arbitrary layers. The material properties, orientations, and layer thicknesses are defined using **Layered Material** node. The offset, and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

Without the Composite Materials Module, only single layer shells and membranes can be modeled. This is still useful, for example for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered Materials in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

It is only possible to select boundaries which are part of the selection of a layered material defined in **Layered Material Link** or **Layered Material Stack** node.

SHELL PROPERTIES

Select the layers in which adiabatic heating is to be modeled.

TABLE 7-63: LAYER SELECTION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the same layer is selected in two nodes being active on the same boundary, the contributions will be exclusive

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

INITIAL VALUES

Enter the **Initial temperature** T_{ini}. The default value is 293.15 K.

THERMODYNAMICS

The density ρ is taken from the material model (Linear Elastic Material or Hyperelastic Material).

The default **Heat capacity at constant pressure** C_{p} uses values **From material**. For **User defined**, enter an expression or value. The default value for the **User defined** is `0` J/(kg·K).

Enter the **Coefficient of adiabatic heating**, β_{ah}. The default value is 1 (dimensionless), which means that dissipative processes contribute 100% as heat sources.

Select the **Dissipative heat source** — **Include all dissipative sources** or **User defined**.

The **Dissipative heat source** list makes it possible to include specific heat sources for the adiabatic heating. Enter a value or expression for the heat source Q_{h} to include. For instance, the dissipated energy density due to creep is available under the variable `shell.Wc` and due to viscoplasticity under the variable `shell.Wvp`. Here `solid` denotes the name of the physics interface node.

TIME STEPPING

The **Backward Euler** method is not available with the Layered Shell interface neither with the Layered Linear Elastic Material nor the Layered Hyperelastic Material in the Shell and Membrane interfaces.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of this field can affect the convergence of the overall solution.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **More** > **Adiabatic Heating**

Ribbon

Physics tab with **Layered Shell** selected:

Boundary > **More** > **Adiabatic Heating**

Rigid Connector

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. The selected shell edges will move as a single rigid object, irrespective of whether they are geometrically adjacent or not.

To add a rigid connector to the top or bottom interface of the shell, use Rigid Connector, Interface.

If the study step is geometrically nonlinear, the rigid connector takes finite rotations into account. The feature is similar to the rigid connectors in the Solid Mechanics and Beam interfaces. Rigid connectors from the Shell, Beam, and Solid Mechanics interfaces can be attached to each other.

You can add functionality to the rigid connector through the following subnodes.

- Thermal Expansion (Rigid Connector) to include thermal expansion.
- Applied Force (Rigid Connector) to apply a force in given point.
- Applied Moment (Rigid Connector) to apply a moment.

- Mass and Moment of Inertia (Rigid Connector) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Connector) to add a translational or rotational spring or damper in a given point.

SHELL PROPERTIES

As the whole edge moves as a single rigid object, it is not possible to select individual layers. The selection in this section only acts as a filter for the boundary selection.

TABLE 7-64: LAYER SELECTIONS; RIGID CONNECTOR.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

All other settings are described in the documentation for Rigid Connector in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Connections** > **Rigid Connector**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Connections** > **Rigid Connector**

Rigid Connector, Interface

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. The selected shell faces will move as a single rigid object, irrespective of whether they are geometrically adjacent or not.

To add a rigid connector to the edge of the shell, use Rigid Connector.

If the study step is geometrically nonlinear, the rigid connector takes finite rotations into account. The feature is similar to the rigid connectors in the Solid Mechanics and Beam interfaces. Rigid connectors from the Shell, Beam, and Solid Mechanics interfaces can be attached to each other.

You can add functionality to the rigid connector through the following subnodes.

- Thermal Expansion (Rigid Connector) to include thermal expansion.
- Applied Force (Rigid Connector) to apply a force in given point.
- Applied Moment (Rigid Connector) to apply a moment.
- Mass and Moment of Inertia (Rigid Connector) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Connector) to add a translational or rotational spring or damper in a given point.

INTERFACE SELECTION

Select the interface or interfaces for which this spring condition is to be added.

TABLE 7-65: INTERFACE SELECTIONS; RIGID CONNECTOR, INTERFACE.

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: **Exterior interfaces**.	When **Selected Interfaces** is selected. Only exterior interfaces can be selected.

- All other settings are described in the documentation for Rigid Connector in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections

Context Menus

Layered Shell > **Face and Domain Constraints, Interfaces** > **Rigid Connector, Interface**

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries > **Interfaces** > **Rigid Connector, Interface**

Thermal Expansion (Rigid Connector)

Add the **Thermal Expansion** subnode to prescribe a deformation of the rigid connector caused by changes in temperature. This makes it possible to reduce stresses caused by the rigid connector being rigid, while there are thermal deformations in the flexible body to which it is attached.

SHELL PROPERTIES

This section is only present when **Thermal Expansion** is used as a subnode to **Rigid Connector**.

Select the layers for which these thermal expansion properties are to be used. You can only select a subset of the layers selected in the parent node.

INTERFACE SELECTION

This section is only present when **Predeformation** is used as a subnode to **Rigid Connector, Interface**.

Select the interfaces for which these thermal expansion properties are to be used. You can only select a subset of the interfaces selected in the parent node

- All other settings are described in the documentation for Thermal Expansion (Rigid Connector) in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Rigid Connector** > **Thermal Expansion**

Layered Shell > **Rigid Connector, Interface** > **Thermal Expansion**

Ribbon

Physics tab with **Rigid Connector** or **Rigid Connector, Interface** node selected in the model tree:

Attributes > **Thermal Expansion**

Attachment

The **Attachment** node is used to define a set of edges on a flexible layered shell which can be used to connect it with other components through a joint in the Multibody Dynamics interface. All the selected edges behave as if they were connected by a common rigid body.

Attachments can be added to boundaries in a Multibody Dynamics or a Solid Mechanics interface, to edges in a Shell or Layered Shell interface, or to points in a

Beam interface. This makes it possible to use a joint in the Multibody Dynamics interface for connecting parts modeled in different physics interfaces.

SHELL PROPERTIES

As the whole edge moves as a single rigid object, it is not possible to select individual layers. The selection in this section only acts as a filter for the boundary selection.

TABLE 7-66: LAYER SELECTIONS; ATTACHMENT.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No

All other settings are described in the documentation for Attachment in the Solid Mechanics interface.

The use of attachments is discussed in the Attachments section in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Connections** > **Attachment**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Connections** > **Attachment**

Thermal Expansion (Attachment)

Add the **Thermal Expansion** subnode to prescribe a deformation of an Attachment caused by changes in temperature. This makes it possible to reduce stresses caused by the attachment being rigid, while there are thermal deformations in the flexible body to which it is attached.

SHELL PROPERTIES

If the parent node has **Use all layers** selected, it is possible to select only a subset. If not, then the layer selection will be the same as in the parent. Since selection of individual

layers is not possible in either case, this section only acts as a filter for the boundary selection.

TABLE 7-67: LAYER SELECTIONS; THERMAL EXPANSION (ATTACHMENT).

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as parent selection when the node is added.	No

- All other settings are described in the documentation for Thermal Expansion (Attachment) in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Attachment** > **Thermal Expansion**

Ribbon

Physics tab with **Attachment** node selected in the model tree:

Attributes > **Thermal Expansion**

Continuity

Add a **Continuity** node to connect two layered shell boundaries sharing a common edge and having different layered materials.

The continuity constraint is added on the common edge in the through-thickness direction.

LAYER SELECTION

Select **Source** and **Destination** layered materials from all the **Layered Material Link** and **Layered Material Stack** available in the component. All the edges are selected by default and the common edges between the two layered materials becomes applicable automatically. Optionally you can remove some of the edges if you do not want to enforce the continuity on all the common edges.

In the **Source**, select the **Layered Materials** based on what you want to use in the continuity constraints. This option is available only when **Layered Material Stack** is chosen.

In the **Destination**, select the **Layered Materials** based on what you want to use in the continuity constraints. In addition to this, an **Offset** can be specified on the destination side which decides the actual connection area between the two layered materials.

You can swap the source and destination by clicking **Swap Source and Destination** ( ).

In order to visualize the connection, click **Layer Cross Section Preview** ( ).

You can use the **Clear All** ( ) and **Reset to Default** ( ) buttons to clear or reset all the selections as well as the offset values in two cases, respectively.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual*:

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

CONNECTION SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Choose a **Connection type** — **Straight** or **Twisted**.

In case the two layered materials have normal orientation in opposite directions, you may want to switch the **Connection type** from **Straight** to **Twisted** in order to connect points in through-thickness direction which are geometrically close to each other.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell > **Continuity**

Ribbon

Physics tab with **Layered Shell** selected:

Edges > **Continuity**

8

Membrane

This chapter describes the **Membrane (mbrn)** interface ( ), which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Membrane Interface

The **Membrane (mbrn)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is mainly used to model prestressed membranes, but can also be used to model a thin cladding on a solid. Membranes can be considered as plane stress elements on boundaries in 3D with a possibility to deform both in the in-plane and out-of-plane directions. There is also a version of the membrane interface for 2D axisymmetric problems. The membrane interface is then applicable to lines since that is what represents boundaries.

The difference between a shell and a membrane is that the membrane does not have any bending stiffness. In most applications, the membrane is used by itself and not as a cladding. A tensile prestress is then necessary in order to avoid singularity because a membrane with no stress or compressive stress has no transverse stiffness. To include the prestress effect, you must enable geometric nonlinearity for the study step.

The Linear Elastic Material is the default material, which adds a linear elastic equation for the displacements and has a Settings window to define the elastic material properties. This material model can also be combined with viscoelasticity.

With the Nonlinear Structural Materials Module, you can also model Nonlinear Elastic and Hyperelastic materials, and add options such as Plasticity, Creep, and Viscoplasticity.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, these default nodes are also added to the **Model Builder**: **Linear Elastic Material**, **Free** (a condition where edges are free, with no loads or constraints), and **Initial Values**. In the case if axial symmetry, an **Axial Symmetry** node is also added. From the **Physics** toolbar, you can then add other nodes that implement, for example, loads and constraints. You can also right-click **Membrane** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern

<name>.<variable_name>. In order to distinguish between variables belonging to different physics interfaces, the name string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is mbrn.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the elastic behavior as quasistatic (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system.

For problems with creep, and sometimes viscoelasticity, the problem can be considered as quasistatic. This is also the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

DISCRETIZATION

In the Membrane interface you can choose not only the order of the discretization, but also the type of shape functions: *Lagrange* or *serendipity*. For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will however give significant reductions of the model size for a given mesh containing quadrilateral elements.

The default is to use **Quadratic serendipity** shape functions for the **Displacement field**.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field u** which has three components (u, v, and w). The name can be changed but the names of fields and dependent variables must be unique within a model.

- Boundary, Edge, Point, and Pair Nodes for the Membrane Interface
- Theory for the Membrane Interface
- Selecting Discretization

Vibrating Membrane: Application Library path
Structural_Mechanics_Module/Verification_Examples/vibrating_membrane

Boundary, Edge, Point, and Pair Nodes for the Membrane Interface

The Membrane Interface has these boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Membrane interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

- The submenus at the **Boundary** level are **Material Models**, **Face and Volume Loads**, **Mass, Spring, and Damper**, and **Face Constraints**.
- The submenus at the **Edge** (3D) or Points (2D axisymmetric) level are **Mass, Spring, and Damper**, **More Constraints**, and **Pairs**.
- For 3D components, there is also a **Points** submenu.

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Additional Creep
- Adiabatic Heating, Layered
- Antisymmetry
- Creep
- Damping
- Edge Load
- External Stress
- Face Load
- Fiber
- Inelastic Strain Rate
- Initial Stress and Strain
- Linear Elastic Material, Layered
- Linear Elastic Material
- Ring Load
- Ring Load, Free
- Safety
- Stabilization
- Symmetry
- Viscoelasticity
- Viscoplasticity
- Wrinkling

These nodes are described for the Solid Mechanics or Shell interfaces:

- Added Mass
- Adhesion
- Adiabatic Heating
- Average Rotation
- Base Excitation
- Body Load[1]
- Contact
- Damping
- Enclosed Cavity
- External Stress–Strain Relation
- Fixed Constraint
- Free
- Friction
- Gravity
- Hygroscopic Swelling
- Hyperelastic Material
- Initial Values
- Linearly Accelerated Frame[1]
- Nonlinear Elastic Material
- Phase
- Point Load
- Point Load, Free
- Point Load (on Axis)
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement[2]
- Prescribed Velocity
- Rotating Frame[1]
- Slip Velocity
- Spring Foundation
- Spring–Damper
- Symmetry Plane
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Fiber)
- Thickness and Offset[3]
- Thickness Change[3]
- Wear

[1] This is selected from the **Face and Volume Loads** submenu for this interface.

[2] At the boundary level, this is selected from the **Face Constraints** submenu for this interface.

[3] Described for the Shell interface.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics Reference Manual*

	In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic membrane and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Inelastic Strain Rate
- Damping
- Wrinkling
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Fiber
- Activation
- Safety

Note: Some options are only available with certain COMSOL products (see www.comsol.com/products/specifications/)

COORDINATE SYSTEM SELECTION

The built in **Boundary System 1** is selected by default. The **Coordinate system** list contains any additional boundary coordinate systems that the model includes. The coordinate system is used for interpreting directions of orthotropic and anisotropic

material data and when stresses or strains are presented in a local system. Many of the possible subnodes inherit the coordinate system settings.

LINEAR ELASTIC MATERIAL

Define the material symmetry and the linear elastic material properties.

Material Symmetry

Select **Material symmetry** — **Isotropic** (the default), **Orthotropic**, or **Anisotropic**. Select:

- **Isotropic** for a linear elastic material that has the same properties in all directions.
- **Orthotropic** for a linear elastic material that has different material properties in orthogonal directions. It is also possible to define **Transversely isotropic** material properties.
- **Anisotropic** for a linear elastic material that has different material properties in different directions.

- Material Models
- Linear Elastic Material
- Orthotropic and Anisotropic Materials

Density

The default **Density** ρ uses values **From material**. For **User defined** enter another value or expression.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

Specification of Elastic Properties for Isotropic Materials

For an **Isotropic** material, from the **Specify** list select a pair of elastic properties for an isotropic material — **Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and**

shear-wave speeds. For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another (see Table 3-1 in the theory chapter).

The individual property parameters are:

- **Young's modulus** (elastic modulus) E.
- **Poisson's ratio** ν.
- **Shear modulus** G.
- **Bulk modulus** K.
- **Lamé parameter** λ and **Lamé parameter** μ.
- **Pressure-wave speed** (longitudinal wave speed) c_p.
- **Shear-wave speed** (transverse wave speed) c_s. This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Specification of Elastic Properties for Orthotropic Materials

When **Orthotropic** is selected from the **Material symmetry** list, the material properties are different in orthogonal directions (principal directions) given by the axes of the selected coordinate system. The **Material data ordering** can be specified in either **Standard** or **Voigt** notation. When **User defined** is selected in 3D, enter three values in the fields for **Young's modulus** $\mathbf{E}$, **Poisson's ratio** ν, and the **Shear modulus** $\mathbf{G}$. The latter defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$

ν_{ij} is defined differently depending on the application field. It is easy to transform among definitions, but check which one the material uses.

You can set an orthotropic material to be **Transversely isotropic**. Then, one principal direction in the material is different from two others that are equivalent. This special

direction is assumed to be the first axis of the selected coordinate system. Because of the symmetry, the following relations hold:

$$E_3 = E_2$$

$$\nu_{13} = \nu_{12}$$

$$G_{23} = \frac{E_2}{2(1+\nu_{23})}$$

$$G_{12} = G_{13}$$

Thus, only five elasticity moduli need to be entered when **User defined** option is selected.

Specification of Elastic Properties for Anisotropic Materials

When **Anisotropic** is selected from the **Material symmetry** list, the material properties vary in all directions. They can be specified using either the **Elasticity matrix**, $\mathbf{D}$ or the **Compliance matrix**, $\mathbf{D}^{-1}$. Both matrices are symmetric. The **Material data ordering** can be specified in either **Standard** or **Voigt** notation. When **User defined** is selected, a 6-by-6 symmetric matrix is displayed.

Mixed Formulation

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which add an extra dependent variable for either the pressure or for the volumetric strain. For details, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, **Strain formulation**, or **Implicit formulation**.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If

the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.
- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

The section is available when you also have the Nonlinear Structural Materials Module. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or **Viscoelasticity**.

DISCRETIZATION

If **Pressure formulation** is used, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**. If **Strain formulation**

is used, select the discretization for the **Auxiliary volumetric strain** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

	The **Discretization** section is available when **Pressure formulation** or **Strain formulation** is selected from the **Use mixed formulation** list. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the integration points for the weak contribution of the feature. Select a method for **Hourglass stabilization** — **Automatic**, **Manual**, or **None** to use in combination with the reduced integration scheme. The default **Automatic** stabilization technique is based on the shape function and shape order of the displacement field.

Control the hourglass stabilization scheme by using the **Manual** option. Select **Shear stabilization** (default) or **Volumetric stabilization**.

When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}. The value should be in the order of magnitude of the equivalent shear modulus.

When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Material Models** > **Linear Elastic Material**

Ribbon

Physics tab with **Membrane** selected:

Boundaries > **Material Models** > **Linear Elastic Material**

Linear Elastic Material, Layered

The **Linear Elastic Material, Layered** node adds the equations for a layered linear elastic membrane.

If the Composite Materials Module analysis is available, this material model can be applied to arbitrary layers in a multilayered membrane. The material properties, orientations, and layer thicknesses are defined using **Layered Material** node. The offset,

and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

Without the Composite Materials Module, only single layer membranes can be modeled. This is still useful, for example for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered Materials in the documentation for the Composite Materials Module.
- See also the discussion in Layered and Nonlayered Membranes.

By adding the following subnodes to the **Linear Elastic Material, Layered** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Inelastic Strain Rate
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Safety

Some of these material models are only available together with the Nonlinear Structural Materials Module (see www.comsol.com/products/specifications/).

SHELL PROPERTIES

For this node, the **Shell Properties** section is only used for selecting a material model, but not individual layers.

TABLE 8-1: LAYER SELECTIONS; LINEAR ELASTIC MATERIAL.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added	No

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

The boundary selection in this node is similar to the **Linear Elastic Material** node. It is however only possible to select boundaries which are part of the selection of a layered material defined in **Layered Material Link** or **Layered Material Stack** node.

LINEAR ELASTIC MATERIAL

Select **Material symmetry** — **Isotropic**, **Orthotropic**, or **Anisotropic** and enter the settings as described for the Linear Elastic Material for the Solid Mechanics interface. If the layers have different types of anisotropy properties, select the one that is most complex.

Note that:

- For **Orthotropic** no values for E_z, ν_{yz}, or ν_{xz} need to be entered due to the membrane assumptions. It is also possible to define **Transversely isotropic** material properties.
- For **User defined Anisotropic** a 6-by-6 symmetric matrix is displayed. Due to the membrane assumptions, you only need to enter values for $D_{11}, D_{12}, D_{22}, D_{14}, D_{24}, D_{55}, D_{66}$, and D_{56}.
- The material orientation is always interpreted in a laminate coordinate system aligned with the membrane boundary together with the orientation of each layer specified on a layered material. The laminate coordinate system is always a **Boundary System**.

For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem. You can then use a mixed formulation, which add an extra dependent variable for either the pressure or for the volumetric strain, see the Mixed Formulation section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, **Strain formulation**, or **Implicit formulation**.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.

- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

The section is available when you also have the Nonlinear Structural Materials Module. Then, to display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or **Viscoelasticity**.

DISCRETIZATION

If **Pressure formulation** is used, select the discretization for the **Auxiliary pressure** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**. If **Strain formulation** is used, select the discretization for the **Auxiliary volumetric strain** — **Automatic**, **Discontinuous Lagrange**, **Continuous**, **Linear**, or **Constant**.

The **Discretization** section is available when **Pressure formulation** or **Strain formulation** is selected from the **Use mixed formulation** list. To display the section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the integration points for the weak contribution of the feature. Select a method for **Hourglass stabilization** — **Automatic**, **Manual**, or **None** to use in combination with the reduced integration scheme. The default **Automatic** stabilization technique is based on the shape function and shape order of the displacement field.

Control the hourglass stabilization scheme by using the **Manual** option. Select **Shear stabilization** (default) or **Volumetric stabilization**.

When **Shear stabilization** is selected, enter a stabilization shear modulus, G_{stb}, and the shear correction factor k_{stb}. The value for G_{stb} should be in the order of magnitude of the equivalent shear modulus.

When **Volumetric stabilization** is selected, enter a stabilization bulk modulus, K_{stb}. The value should be in the order of magnitude of the equivalent bulk modulus.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Material Models** > **Linear Elastic Material, Layered**

Ribbon

Physics tab with **Membrane** selected:

Boundaries > **Material Models** > **Linear Elastic Material, Layered**

Viscoelasticity

Use the **Viscoelasticity** subnode to add viscous stress contributions to a material model. Viscoelasticity can be added to Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material.

If the Composite Materials Module is available, the viscoelasticity model can be applied to arbitrary layers in a multilayered membrane.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Viscoelasticity** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this viscoelastic model is to be used.

TABLE 8-2: LAYER SELECTIONS; VISCOELASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected.

If the viscoelastic model differs between layers, you will need to add several **Viscoelasticity** nodes with different layer selections.

If the same layer is selected in two **Viscoelasticity** nodes being active on the boundary, then there will be two viscoelastic contributions.

- All other settings for the **Viscoelasticity** node are described in the documentation for Viscoelasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Viscoelasticity**

Membrane > **Linear Elastic Material, Layered** > **Viscoelasticity**

Membrane > **Nonlinear Elastic Material** > **Viscoelasticity**

Membrane > **Hyperelastic Material** > **Viscoelasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Viscoelasticity**

Plasticity

Use the **Plasticity** subnode to define the properties for modeling elastoplastic materials. This material model can be used together with a Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, or a Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the plasticity model can be applied to arbitrary layers in a multilayered membrane.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Plasticity** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this plasticity model is to be used.

TABLE 8-3: LAYER SELECTIONS; PLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the plasticity model differs between layers, you will need to add several **Plasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Plasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If there are two **Plasticity** nodes where the same layer is selected on the same boundary, the latter will be overridden on the common selections.

- All other settings for the **Plasticity** node are described in the documentation for Plasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Plasticity**
Membrane > **Linear Elastic Material, Layered** > **Plasticity**
Membrane > **Nonlinear Elastic Material** > **Plasticity**
Membrane > **Hyperelastic Material** > **Plasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Plasticity**

Set Variables

Use the **Set Variables** subnode to Plasticity of a Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, or Hyperelastic Material node to reset plasticity variables according to a **Setting condition** that you define. When the **Setting condition** is satisfied, the plasticity variables are reset to the specified values.

SET VARIABLES

Enter the **Setting condition**. This is a Boolean expression that will determine when the plastic variables are reset.

From the **Equivalent plastic strain** list, select **Do not set** or **User defined**. The default **User defined** value is zero.

From the **Plastic strain tensor** list, select **Do not set** or **User defined**. The default **User defined** values are zero for all components of the **Plastic strain tensor**.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Plasticity** > **Set Variables**
Membrane > **Linear Elastic Material, Layered** > **Plasticity** > **Set Variables**
Membrane > **Nonlinear Elastic Material** > **Plasticity** > **Set Variables**
Membrane > **Hyperelastic Material** > **Plasticity** > **Set Variables**

Ribbon

Physics tab with **Plasticity** node selected in the model tree:

Attributes > **Set Variables**

Creep

Use the **Creep** subnode to define the creep properties of the material model. Creep can be added to Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, creep models can be applied to arbitrary layers in a multilayered membrane.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Creep** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this creep model is to be used.

TABLE 8-4: LAYER SELECTIONS; CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Add several **Creep** nodes with different layer selections when the creep model differs between layers. If the model is the same, and only the material data differ, you can use a single **Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

- All other settings for the **Creep** node are described in the documentation for Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Creep**
Membrane > **Linear Elastic Material, Layered** > **Creep**
Membrane > **Nonlinear Elastic Material** > **Creep**
Membrane > **Hyperelastic Material** > **Creep**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Creep**

Additional Creep

Use the **Additional Creep** subnode to define additional contributions to the creep model defined by the parent Creep node, such as primary or tertiary creep behavior. A **Creep**

node can have any number of **Additional Creep** subnodes with different settings to model advanced creep mechanisms.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, creep models can be applied to arbitrary layers in a multilayered membrane.

See also Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Additional Creep** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this creep model is to be used.

TABLE 8-5: LAYER SELECTIONS; CREEP.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the creep model differs between layers, you will need to add several **Additional Creep** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Additional Creep** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Additional Creep** nodes being active on the boundary, then there will be two creep strain contributions.

- All other settings for the **Additional Creep** node are described in the documentation for Additional Creep in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Creep** > **Additional Creep**
Membrane > **Linear Elastic Material, Layered** > **Creep** > **Additional Creep**
Membrane > **Nonlinear Elastic Material** > **Creep** > **Additional Creep**
Membrane > **Hyperelastic Material** > **Creep** > **Additional Creep**

Ribbon

Physics tab with **Creep** node selected in the model tree:

Attributes > **Additional Creep**

Viscoplasticity

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model. Viscoplasticity can be added to Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material.

The Nonlinear Structural Materials Module is required for this feature. For details, see www.comsol.com/products/specifications/.

If the Composite Materials Module is available, the viscoplastic model can be applied to arbitrary layers in a multilayered membrane.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present when the **Viscoplasticity** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this plasticity model is to be used.

TABLE 8-6: LAYER SELECTIONS; VISCOPLASTICITY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the viscoplasticity model differs between layers, you will need to add several **Viscoplasticity** nodes with different layer selections. If the model is the same, and only the material data values differ, you can use a single **Viscoplasticity** node where **From material** is used to define the values. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Viscoplasticity** nodes being active on the same boundary, the second definition will override the previous.

- All other settings for the **Viscoplasticity** node are described in the documentation for Viscoplasticity in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide material parameters with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Viscoplasticity**
Membrane > **Linear Elastic Material, Layered** > **Viscoplasticity**
Membrane > **Nonlinear Elastic Material** > **Viscoplasticity**
Membrane > **Hyperelastic Material** > **Viscoplasticity**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Viscoplasticity**

Wrinkling

Use the **Wrinkling** node to define the settings for the Newton–Raphson method that solves the wrinkling equations. Wrinkling can be added to a Linear Elastic Material or to a Hyperelastic Material.

Select a **Termination criterion for local method** — **Step size**, **Residual**, or **Step size or residual**. Set the **Maximum number of local iterations** in the Newton–Raphson loop when solving the local wrinkling equations. The defaults value is 50 local iterations.

- When **Step size** is chosen as termination criterion, set the **Absolute tolerance** and the **Relative tolerance** to check the convergence of the local wrinkling equations based on the step size in the Newton–Raphson loop. The final tolerance is computed based on the current solution of the local variable and the entered values.
- When **Residual** is chosen as termination criterion, set the **Residual tolerance** to check the convergence of the local wrinkling equations based on the residual of each equation.
- When **Step size or residual** is chosen, it is sufficient that one of the conditions is fulfilled. Setting either the **Absolute tolerance** and **Relative tolerance** or the **Residual tolerance** to zero ignores the corresponding convergence check. An error is returned if all tolerances are set to zero.

The theory is described under Wrinkling in the Theory for the Membrane Interface section.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Wrinkling**
Membrane > **Hyperelastic Material** > **Wrinkling**

Ribbon

Physics tab with **Linear Elastic Material** or **Hyperelastic Material** node selected in the model tree:

Attributes > **Wrinkling**

Fiber

Add one or more **Fiber** nodes to add stiffness in specified directions to a membrane. The **Fiber** feature can be used together with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material**. The assumption is that the volume of the fibers is small when compared to the base material.

You can include the effect of thermal expansion of the fiber by adding a **Thermal Expansion** subnode (see Thermal Expansion (Fiber)).

FIBER MODEL

Linear Elastic Material or Nonlinear Elastic Material

Select a **Material Model** — **Linear elastic** or **Uniaxial data**.

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- For **Linear elastic**, enter the **Young's modulus** E_{fiber} and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions.
- For **Uniaxial data**, enter the **Uniaxial stress function** $\sigma_{\text{ax,a}}$ and **Density** ρ_{fiber}. The defaults are taken **From material**. Select **User defined** to enter other values or expressions. The default user defined expression for $\sigma_{\text{ax,a}}$ is the linear function `210[GPa]*<item>.eea`, where the variable `<item>.eea` is the elastic strain in the fiber direction, and `<item>` corresponds to the tag of the **Fiber** node, for example `mbrn.lemm1.fib1`. See Fibers for Linear and Nonlinear Elastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want

this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

Hyperelastic Material

Select a **Material Model** — **Holzapfel–Gasser–Ogden**, **Linear elastic**, **Uniaxial data**, or **User defined**.

From the **Material** list, select the **Boundary material** (the default) or any other material to define the fiber's properties. In most cases, you would use the boundary material for the base material, and additional **Material** nodes without boundary selection as the fiber material. This selection will be overridden if you use the **User defined** option for a certain material property.

- The **Holzapfel–Gasser–Ogden** model defines an exponential strain energy density W_{fiber}. The values for the **Fiber stiffness** k_1, **Model parameter** k_2, and **Fiber dispersion** k_3 are as a default taken **From material**. Select **User defined** to enter other values or expressions.
- Select **Linear elastic** or **Uniaxial data** to describe the fiber stiffness. See Linear Elastic Material or Nonlinear Elastic Material for details.
- If **User defined** is selected from the list, select a **Material symmetry** — **Isotropic**, **Transversely isotropic**, or **Anisotropic**.

  Depending on the choice, different stretch invariants are available to define the **Fiber strain energy density** W_{fiber}. The default expression for W_{fiber} is the function `0[Pa]*(<item>.IaCIe-1)`, where the variable `<item>.IaCIe` is the isochoric stretch invariant in the fiber direction, and `<item>` corresponds to the tags of the **Fiber** node, for example `mbrn.hmm1.fib1`. See Fibers for Hyperelastic Materials in the *Structural Mechanics Theory* chapter for details and more options.

For the **Holzapfel–Gasser–Ogden** model, select the **Use isochoric deformation** checkbox to account for the fiber incompressibility. See Compressible Fibers in the *Structural Mechanics Theory* chapter for details.

Select the **Stiffness in tension only** checkbox in order to make the fibers contribute to the total stiffness only when the strain in the fibers is tensile.

Select the **Contribute to total stress** checkbox if the stress in the fibers should be added to the stress tensor of the parent material in an average sense. Usually, you do not want this, since the fiber stress would then affect other material options in the matrix material, such as plasticity.

DISTRIBUTION AND ORIENTATION

From the **Specify** list select how the fiber volume fraction is entered — **Volume distribution** or **Surface distribution**.

- For **Volume distribution**, enter the fiber **Volume fraction** v_{fiber}. To be consistent with the underlying assumptions, it should not exceed a few percent.
- For **Surface distribution**, the fibers are placed at the midsurface. Enter the fiber **Diameter** d_{fiber} and the fiber **Spacing** s_{fiber} to compute the fiber volume fraction. To be consistent with the underlying assumptions, the diameter–spacing ratio should not exceed a few percent.

Select the **Fiber orientation** from the list. The available choices are the axis directions of the coordinate system selected in the **Coordinate System Selection** section, or **User defined**. For **User defined**, enter a **Direction**. The direction vector $\mathbf{a}$ is interpreted in the in-plane directions of the selected coordinate system.

See also Distributed Fiber Models in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Fiber**
Membrane > **Nonlinear Elastic Material** > **Fiber**
Membrane > **Hyperelastic Material** > **Fiber**

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Fiber**

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

Thermal expansion can be modeled for Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material. For **Linear Elastic**

Material, Layered, the thermal expansion can be applied to arbitrary layers in a multilayered membrane when the Composite Materials Module is available.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Thermal Expansion** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this thermal strain contribution is to be added.

TABLE 8-7: LAYER SELECTIONS; THERMAL EXPANSION.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected.

If the **Input type** differs between layers, you will need to add several **Thermal Expansion** nodes with different layer selections. If only the material data values differ, you can use a single **Thermal Expansion** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Thermal Expansion** nodes being active on the same boundary, the second definition will override the previous.

Thermal strains are proportional to the temperature, while structural strains are computed from the gradient of the displacement field. It is good practice to match the discretization order of thermal and structural strains.

When adding a **Thermal Expansion** subnode, and the temperature field is computed by another physics interface (often the Heat Transfer in Shells interface); use a discretization one order lower for the temperature field than what is used for the displacement field.

- All other settings for the **Thermal Expansion** node are described in the documentation for Thermal Expansion (for Materials) in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the expansion with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Thermal Expansion**
Membrane > **Linear Elastic Material, Layered** > **Thermal Expansion**
Membrane > **Nonlinear Elastic Material** > **Thermal Expansion**
Membrane > **Hyperelastic Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Use the **Hygroscopic Swelling** subnode to add an internal thermal strain caused by changes in temperature.

Hygroscopic swelling can be added to Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material. For **Linear Elastic Material, Layered**, swelling can be applied to arbitrary layers in a multilayered membrane when the Composite Materials Module is available.

See also the discussion in Layered and Nonlayered Membranes.

SHELL PROPERTIES

This section is present only when the **Hygroscopic Swelling** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this swelling strain contribution is to be added.

TABLE 8-8: LAYER SELECTIONS; HYGROSCOPIC SWELLING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected.

If the **Moisture concentration type** differs between layers, you will need to add several **Hygroscopic Swelling** nodes with different layer selections. If only the material data values differ, you can use a single **Hygroscopic Swelling** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Hygroscopic Swelling** nodes being active on the same boundary, the second definition will override the previous.

Hygroscopic strains are proportional to the concentration, while structural strains are computed from the gradient of the displacement field. It is good practice to match the discretization order of hygroscopic and structural strains.

When adding a **Hygroscopic Swelling** subnode, and the concentration field is computed by another physics interface; use a discretization one order lower for the concentration field than what is used for the displacement field.

- All other settings for the **Hygroscopic Swelling** node are described in the documentation for Hygroscopic Swelling in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can provide parameters for the expansion with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Hygroscopic Swelling**
Membrane > **Linear Elastic Material, Layered** > **Hygroscopic Swelling**
Membrane > **Nonlinear Elastic Material** > **Hygroscopic Swelling**
Membrane > **Hyperelastic Material** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to the **Linear Elastic Material**, in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense, but rather a contribution to the constitutive relation.

Initial stresses and strains can be included with Linear Elastic Material, Linear Elastic Material, Layered, and Nonlinear Elastic Material. For **Linear Elastic Material, Layered**, the contribution can be applied to arbitrary layers in a multilayered membrane when the Composite Materials Module is available

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

SHELL PROPERTIES

This section is present only when the **Initial Stress and Strain** node is used under Linear Elastic Material, Layered.

Select the layer or layers for which this stress or strain contribution is to be added.

TABLE 8-9: LAYER SELECTIONS; INITIAL STRESS AND STRAIN.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the initial stress or strain differ between layers, you will need to add several **Initial Stress and Strain** nodes with different layer selections.

If the same layer is selected in two **Initial Stress and Strain** nodes being active on the same boundary, the contributions will be added.

- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe stress or strain values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

COORDINATE SYSTEM SELECTION

The given initial stresses and strains are interpreted in this system.

For details about how local coordinate systems are used in the Membrane interface, see Local Coordinate Systems.

INITIAL STRESS AND STRAIN

Specify the initial stress as the **Initial local in-plane force** N_0 and the initial strain as the **Initial local in-plane strain** ε_0 (dimensionless). In addition, you can also provide the **Initial out-of-plane strain** ε_{n0}.

If you know the stress, rather than the force per reference length, type in the stress multiplied by the membrane thickness `mbrn.d`. In a geometrically nonlinear analysis, the stresses should be interpreted as Second Piola–Kirchhoff stresses, and the strains should be interpreted as Green–Lagrange strains.

- For details about initial stresses and strains, see Inelastic Strain Contributions and Initial Stress and Strain.
- For details about the different strain measures, see Deformation Measures.
- For details about the different stress measures, see Defining Stress.

Prestressed Micromirror: Application Library path **MEMS_Module/Actuators/micromirror**

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Initial Stress and Strain**
Membrane > **Linear Elastic Material, Layered** > **Initial Stress and Strain**
Membrane > **Nonlinear Elastic Material** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, or **Nonlinear Elastic Material** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode in order to specify an additional stress contribution which is not part of the constitutive relation. The external stress can be added to the total stress tensor, or act only as an extra load contribution.

External stresses can be added to Linear Elastic Material, Linear Elastic Material, Layered, Nonlinear Elastic Material, and Hyperelastic Material. For **Linear Elastic Material, Layered**, the contribution can be applied to arbitrary layers in a multilayered membrane when the Composite Materials Module is available.

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor, whereas the option to use the stress as only a load contribution exists only in **External Stress**.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** node. Select the layers in which initial stress and strain needs to be modeled.

TABLE 8-10: LAYER SELECTIONS; EXTERNAL STRESS.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the external stress input type differs between layers, you will need to add several **External Stress** nodes with different layer selections.

If the same layer is selected in two **External Stress** nodes being active on the same boundary, the contributions will be added.

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor (Material)**, **Stress tensor (Spatial)**, or **In-plane force**.

- When **Stress tensor (Material)** is selected, you enter the external stress in the form of a Second Piola–Kirchhoff stress tensor. The **External stress tensor** list will contain all stress tensors announced by any physics interface, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** S_{ext} as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** to determine the effect of the contribution. Stress tensor components which are not in the plane of the membrane will be ignored.
- When **Stress tensor (Spatial)** is selected, you enter the external stress in the form of Cauchy stress tensor. The components are interpreted in the selected coordinate system. Depending on the properties of the tensor, you can enter the data for the **External stress tensor** σ_{ext} as **Isotropic**, **Diagonal**, or **Symmetric**. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** to determine

the effect of the contribution. Stress tensor components which are not in the plane of the membrane will be ignored.

- When **In-plane force** is selected, you enter the external stress in the form of section forces. Specify the initial stress as values or expressions for the **In-plane force** N_{ext}, which has components in the plane of the membrane, along the directions given by the selected local coordinate system. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** to determine the effect of the contribution.

Since all stress tensor representations coincide in a geometrically linear analysis, **Stress tensor (Spatial)** is needed only in the case of a geometrically nonlinear analysis. The stress tensor is entered using a Cauchy stress tensor representation, and is internally transformed to a Second-Piola stress tensor.

- For theory, see External Stress.
- For details about the different stress measures, see Defining Stress.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **External Stress**
Membrane > **Linear Elastic Material, Layered** > **External Stress**
Membrane > **Nonlinear Elastic Material** > **External Stress**
Membrane > **Hyperelastic Material** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **External Stress**

Inelastic Strain Rate

The **Inelastic Strain Rate** subnode allows you to provide inelastic strain contributions to the material models **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** on a variety of formats as in External Strain. Here, however, the inelastic contribution is given as a rate, and the total inelastic

contribution is computed by integrating this rate in time. For **Linear Elastic Material, Layered**, the inelastic contribution can be applied to arbitrary layers in a multilayered membrane when the Composite Materials Module is available.

SHELL PROPERTIES

See the documentation for the Inelastic Strain Rate node in the Layered Shell chapter.

All other settings for the **Inelastic Strain Rate** node are described in the documentation for Inelastic Strain Rate in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Inelastic Strain Rate**
Membrane > **Linear Elastic Material, Layered** > **Inelastic Strain Rate**
Membrane > **Nonlinear Elastic Material** > **Inelastic Strain Rate**
Membrane > **Hyperelastic Material** > **Inelastic Strain Rate**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Inelastic Strain Rate**

Safety

Use the **Safety** subnode to set up variables which can be used to check the risk of failure according to various criteria. It can be used in combination with Linear Elastic Material, Linear Elastic Material, Layered, and Nonlinear Elastic Material. Four different variables describing the failure risk are defined, as described in Table 8-11.

You can add any number of **Safety** nodes to a single material model. The contents of this feature do not affect the analysis results as such, so you can add **Safety** nodes after

having performed an analysis, and just do an **Update Solution** in order to access to the new variables for result evaluation.

TABLE 8-11: VARIABLES FOR SAFETY FACTOR EVALUATION.

VARIABLE	DESCRIPTION	CRITERION FULFILLED	CRITERION VIOLATED
Failure index, FI	For a linear criterion, this is the ratio between the computed value and the given limit.	FI<1	FI>1
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=1
Safety factor, SF	For a linear criterion, this is 1/FI.	SF>1	SF<1
Margin of safety, MoS	SF-1	MoS>0	MoS<0

For orthotropic and anisotropic failure criteria, the directions are given by the coordinate system selection in the parent node.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** node.

Select the layer or layers for which the safety factor variables are to be defined.

TABLE 8-12: LAYER SELECTIONS; SAFETY.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

Each **Safety** subnode defines its own set of variables, so there is no interaction if you add several such nodes with the same selection.

- All other settings for the **Safety** node are described in the documentation for Safety in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe parameters for the allowable values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Variables** > **Safety**
Membrane > **Linear Elastic Material, Layered** > **Variables** > **Safety**
Membrane > **Nonlinear Elastic Material** > **Variables** > **Safety**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, or **Nonlinear Elastic Material** node selected in the model builder tree:

Attributes > **Variables** > **Safety**

Damping

Use the **Damping** subnode to add several types of damping to the material model. Damping can be used in Time Dependent, Eigenfrequency, and Frequency Domain studies; for other study types the settings in the **Damping** subnode are ignored.

SHELL PROPERTIES

This section is only present when this node is added under a **Linear Elastic Material, Layered** node.

Select the layer or layers for which this damping model is to be used.

TABLE 8-13: LAYER SELECTIONS; DAMPING.

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

When the damping model differs between layers, you will need to add several **Damping** nodes with different layer selections. If the model is the same, and only the material data values differ, you can for most models use a single **Damping** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

- All other settings for the **Damping** node are described in the documentation for Damping in the Solid Mechanics interface.
- For a general description of layer and interface selections, see The Shell Properties and Interface Selection Sections.
- You can describe damping values with a through-thickness variation by explicitly or implicitly using expressions containing the extra dimension coordinate as described in Using the Extra Dimension Coordinates.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Linear Elastic Material** > **Damping**
Membrane > **Linear Elastic Material, Layered** > **Damping**
Membrane > **Nonlinear Elastic Material** > **Damping**
Membrane > **Hyperelastic Material** > **Damping**

Ribbon

Physics tab with **Linear Elastic Material**, **Linear Elastic Material, Layered**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

Attributes > **Damping**

This node is available only in 3D. When the model is axisymmetric, use the Symmetry Plane node instead.

The **Symmetry** node adds a boundary condition that represents symmetry in the geometry and in the loads. A symmetry condition is free in the plane and fixed in the out-of-plane direction. The symmetry plane is formed by the normal to the membrane surface and the edge tangent.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying a symmetry edge. From the **Coordinate system** list select from:

- **Local edge system** (the default).
- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This section is only shown if the **Local Edge System** is chosen in the **Coordinate System Selection**. If symmetry is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_n. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.mbrn.sym1.un`.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **More Constraints** > **Symmetry**

Ribbon

Physics tab with **Membrane** selected:

Edges > **More Constraints** > **Symmetry**

Antisymmetry

This node is available only in 3D.

The **Antisymmetry** node adds a boundary condition for an antisymmetry boundary. An antisymmetry condition is fixed in the plane and free in the out-of-plane direction. The antisymmetry plane is formed by the normal to the boundary and the edge tangent.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying a symmetry edge. From the **Coordinate system** list select from:

- **Local edge system** (the default).
- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate system.

For details about the definition of local edge systems, see Local Edge System.

FACE DEFINING THE LOCAL ORIENTATIONS

This section is only shown if the **Local Edge System** is chosen in the **Coordinate System Selection**. If symmetry is prescribed for an edge which is shared between boundaries,

the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

ANTISYMMETRY

If another coordinate system than the **Local edge system** is used, select an **Axis to use as symmetry plane normal**. Select **1**, **2**, or **3** for the first, second, or third axis, respectively.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **More Constraints** > **Antisymmetry**

Ribbon

Physics tab with **Membrane** selected:

Edges > **More Constraints** > **Antisymmetry**

Stabilization

Membranes rely on the stress-stiffening effect to be stable. This implies that membranes are unstable when not prestressed with in-plane loads. There are

nonetheless many situations in which the value of the prestress is unknown, and a preliminary loading analysis is performed to find it. Add a **Stabilization** node to include fictitious stabilizing forces during the initial loading steps, when the load associated to the prestress is close to zero.

QUASISTATIC STABILIZATION

The stabilization works by connecting spring foundations for the geometric entities included in the selection. COMSOL Multiphysics makes an estimation of the spring stiffness k_{char} required to stabilize the problem. If this estimate of the stiffness is too small or too large, enter a new value for the **Stiffness multiplier** f_{stb}.

It is essential that the stabilization force are relaxed once the membrane load is sufficiently high to make the problem non singular and the stress-stiffening effect is established. Select a **Method** — **Automatic** or **Manual**, for how to relax the stabilization forces.

For the **Manual** method, select **Active for** — **First iteration**, **First step**, **Specified number of steps**, or **User-defined condition**, to control how the stabilization forces are relaxed.

- **First iteration** applies the stabilization forces for the first iteration of the nonlinear solver only.
- **First step** applies the stabilization forces for the first parameter step or time step. Enable **Iteration relaxation** to relax the spring stiffness with each iteration of the nonlinear solver.
- For **Specified number of steps**, enter the number of parameter steps or time steps for which the stabilization forces are active. By enabling the **Step relaxation**, the spring stiffness is reduced with each step number. If **Iteration relaxation** is enabled, the spring stiffness is relaxed with each nonlinear iteration of the final specified step number.

For **User-defined condition**, enter a Boolean expression. The stabilization forces are active when the Boolean expression is evaluated to true. This option can be useful to control the activation based on a specific parameter value in a continuation sweep, or at a specific time in a time-dependent analysis.

To monitor the stabilization energy added, add a **Global Variable Probe** of the variable `<phys>.<stabilization_feature>.Wstb`, and output probe values while solving in the **Fully Coupled** or **Segregated** solver. This setting allows to track how the stabilization scheme relaxes with the nonlinear iterations.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **More** > **Stabilization**

Ribbon

Physics tab with **Membrane** selected:

Boundary > **More** > **Stabilization**

Adiabatic Heating, Layered

The **Adiabatic Heating, Layered** node adds the equations for adiabatic heating in layered membranes caused by abrupt changes in temperature due to fast deformation.

The increase in temperature is then defined by solving the distributed equation

$$\rho C_p \dot{T} = \beta_{ah} Q_h$$

where ρ is the density, C_p is the heat capacity at constant pressure, T is the temperature field, β_{ah} is the coefficient of adiabatic heating, and Q_h corresponds to the heat sources due to mechanical dissipative processes.

Adiabatic heating is only available with some COMSOL products (see www.comsol.com/products/specifications/).

If the Composite Materials Module is available, adiabatic heating can be applied to arbitrary layers. The material properties, orientations, and layer thicknesses are defined using **Layered Material** node. The offset, and local coordinate system, in which material orientations and results are interpreted, is defined by **Layered Material Link** or **Layered Material Stack** node.

Without the Composite Materials Module, only single layer shells and membranes can be modeled. This is still useful, for example for some multiphysics couplings. For single layer materials, an ordinary **Material** node can be used, as long you include a **Shell** property group in which, for example, the thickness is given.

- For a general description about layered materials, see Layered Materials in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

It is only possible to select boundaries which are part of the selection of a layered material defined in **Layered Material Link** or **Layered Material Stack** node.

SHELL PROPERTIES

Select the layers in which adiabatic heating is to be modeled.

TABLE 8-14: LAYER SELECTION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When **Use all layers** is not selected. Only a subset of the layers selected in the parent can be selected.

If the same layer is selected in two nodes being active on the same boundary, the contributions will be exclusive

For a general description of this section, see Layer and Interface Selections in the documentation for the Composite Materials Module.

INITIAL VALUES

Enter the **Initial temperature** T_{ini}. The default value is 293.15 K.

THERMODYNAMICS

The density ρ is taken from the material model (Linear Elastic Material or Hyperelastic Material).

The default **Heat capacity at constant pressure** C_{p} uses values **From material**. For **User defined**, enter an expression or value. The default value for the **User defined** is `0` J/(kg·K).

Enter the **Coefficient of adiabatic heating**, β_{ah}. The default value is 1 (dimensionless), which means that dissipative processes contribute 100% as heat sources.

Select the **Dissipative heat source** — **Include all dissipative sources** or **User defined**.

The **Dissipative heat source** list makes it possible to include specific heat sources for the adiabatic heating. Enter a value or expression for the heat source Q_{h} to include. For instance, the dissipated energy density due to creep is available under the variable `shell.Wc` and due to viscoplasticity under the variable `shell.Wvp`. Here `solid` denotes the name of the physics interface node.

TIME STEPPING

The **Backward Euler** method is not available with the Layered Shell interface neither with the Layered Linear Elastic Material nor the Layered Hyperelastic Material in the Shell and Membrane interfaces.

Domain ODEs

No settings are needed for the **Domain ODEs** method. However, this method adds degrees-of-freedom that are solved as part of the general solver sequence. The scaling of this field can affect the convergence of the overall solution.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **More** > **Adiabatic Heating, Layered**

Ribbon

Physics tab with **Membrane** selected:

Boundary > **More** > **Adiabatic Heating, Layered**

Face Load

Add a **Face Load** node to apply tractions or pressure on boundaries where **Membrane** is active.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems and in the *COMSOL Multiphysics Reference Manual.*

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Total force**, or **Pressure**. Then enter values or expressions based on the selection and the space dimension.

- For **Force per reference area** and **Force per deformed area**, the traction components are given explicitly.
- For **Total force**, COMSOL Multiphysics divides the total force by the area of the boundaries where the load is active. When working with curved boundaries or local coordinate systems, use this option carefully, as the result is not always intuitive.

- For **Pressure**, a scalar input is given, and the orientation of the load is given by the normal to the boundary. The pressure is positive when directed toward the membrane. In a geometrically nonlinear analysis, the current surface normal and area are used.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference area	$\mathbf{f}_A$	N/m^2	boundaries	3D (x, y, z) 2D axisymmetric (r, z)
Force per deformed area	$\mathbf{f}_a$	N/m^2	boundaries	3D (x, y, z) 2D axisymmetric (r, z)
Total force	$\mathbf{F}_{tot}$	N	boundaries	3D (x, y, z) 2D axisymmetric (r, z)
Pressure	p	Pa	boundaries	3D 2D axisymmetric

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

- Modeling Thin Layers in the *Structural Mechanics Modeling* chapter.
- Thin Layers in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Face and Volume Loads** > **Face Load**

Ribbon

Physics tab with **Membrane** selected in the model tree:

Boundaries > **Face and Volume Loads** > **Face Load**

Edge Load

Add an **Edge Load** to apply a force distributed along an edge.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems and in the *COMSOL Multiphysics Reference Manual.*

FACE DEFINING THE LOCAL ORIENTATIONS

If the load is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

FORCE

Select a **Load type** — **Force per reference area**, **Force per deformed area**, **Force per reference length**, **Force per deformed length**, or **Total force**. Then enter values or expressions for the components in based on the selection:

- The load per reference unit area $\mathbf{f}_A$.
- The load per deformed unit area $\mathbf{f}_a$.
- The load per reference unit length $\mathbf{f}_L$.
- The load per deformed unit length $\mathbf{f}_l$.
- The total force $\mathbf{F}_{tot}$. COMSOL Multiphysics then divides the total force by the length of the edge where the load is active, and then the edge load is applied in the same way as in **Force per reference length**.

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Edge Load**

Ribbon

Physics tab with **Membrane** selected:

Edges > **Membrane** > **Edge Load**

Ring Load

A load applied to a point, not located at the axis of revolution, in a 2D axisymmetric model actually represents a ring load. Add a **Ring Load** to points located at $R > 0$ to model such loads.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems and in the *COMSOL Multiphysics Reference Manual*.

FACE DEFINING THE LOCAL ORIENTATIONS

This section is not shown if the **Global coordinate system** is chosen in the **Coordinate System Selection**. If the load is prescribed for an edge which is shared between boundaries, the edge system can be ambiguous. Select the boundary which should define the edge system. The default is **Use face with lowest number**.

FORCE

Select the **Load type** — **Force per reference area**, **Force per deformed area**, **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for $\mathbf{f}_A$, $\mathbf{f}_a$, $\mathbf{f}_L$, $\mathbf{f}_l$, or $\mathbf{F}_{tot}$.

- The **Load** list normally only contains **User defined**. When combining with another physics interface that can provide this type of load, it is also possible to choose a predefined load from this list.
- The **Total force** option should not be directly interpreted as a resultant, since the orientation is changing. Rather, it should be interpreted as the per reference length values, multiplied by $2\pi r$, where r is the radius of the point.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Ring Load**

Ribbon
Physics tab with **Membrane** selected:

Points > **Membrane** > **Ring Load**

Ring Load, Free

A load applied to a point, not located at the axis of revolution, in a 2D axisymmetric model actually represents a ring load. Add a **Ring Load, Free** node to describe concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Imported meshes, where there may not be suitable points for load application
- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable `<phys>.<load_tag>.num_ignored` (for example, `membrane.rlf1.num_ignored`).
- The local stress field in the element where a point load acts will not be reliable.

LOCATION AND FORCE

Select a **Frame** — **Material** or **Spatial** that determines how the given coordinates are interpreted in case of a geometrically nonlinear analysis. When **Material** is selected, the load location on the body is interpreted with respect to the initial configuration. When **Spatial** is selected, then the load location is given in space fixed coordinates, which usually means that the material point on the body where the load is applied will change with deformation even if the coordinate values are constant.

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**R**, **Z**). If spatial frame input is used, then the spatial coordinates (**r**, **z**) are used. Then, enter the force values, **FxI** and **FyI**. The force vector is interpreted in the coordinate system selected in the **Coordinate System Selection** section.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Membrane > **Ring Load, Free**

Ribbon

Physics tab with **Membrane** selected:

Global > **Membrane** > **Ring Load, Free**

9

Beam

This chapter describes the Beam interface ( ), which you find under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Beam Interface

The **Beam (beam)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for modeling slender structural elements, having a significant bending stiffness. The formulation allows geometric nonlinearity, with large rotations and small strains, and beams can be modeled on 2D boundaries and 3D edges.

Two-noded straight elements with a Hermitian formulation are used. Two different assumptions about the physics can be used:

- Euler (or Euler-Bernoulli) theory. This formulation is intended for slender beams, and do not take shear deformations into account.
- Timoshenko theory. In this formulation that extends the beam theory to 'thick' beams, shear deformations are taken into account. In a dynamic analysis, inertial effects from rotation are also included.

Among the computed results are displacements, rotations, stresses, strains, and section forces. In addition to giving the beam properties explicitly in terms of area, moment of inertia, and so on, several predefined common cross-section types are available. Cross-section data to be used in **Cross-Section Data** settings can be computed using The Beam Cross Section Interface.

The Linear Elastic Material node is the only available material model.

When this physics interface is added, these default nodes are also added to the Model Builder: **Linear Elastic Material**, **Cross-Section Data**, **Free** (a condition where points are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, loads and constraints. You can also right-click **Beam** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `beam`.

SKETCH

In the **Sketch** section, a conceptual sketch of the degrees of freedom in the Beam interface is shown.

BEAM FORMULATION

Select **Euler-Bernoulli** or **Timoshenko** to use the appropriate beam theory. Timoshenko theory includes the effects of shear flexibility and rotary inertia, and is appropriate for beams with cross-section dimensions which are large relative to the length of the beam.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

This is often the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

AUTOMATED MODEL SETUP

This section will only be displayed if a mesh on NASTRAN® format, containing RBE2 elements, has been imported in an **Import** node under **Mesh**. The purpose is to automatically create rigid connectors from RBE2 elements in the NASTRAN file.

An RBE2 element represents a rigid connection between a set of mesh vertices. This means that it can, and often does, connect elements from different physics interfaces.

In the menu in the section title, you can select **Create Rigid Connectors from RBE2**. The effect is that one rigid connector will be created for each RBE2 element in the imported file. This will happen for all physics interfaces in the **Interfaces** list. Supported interfaces are: Solid Mechanics, Shell, Beam, and Multibody Dynamics. If there are RBE2 elements spanning more than one physics interface, they will be automatically connected.

The created rigid connectors will have point, edge, and boundary selections as inferred from the nodes in the RBE2 element and the mesh connectivity. The ‘independent node’ of the RBE2 element is used as center of rotation for the rigid connector.

The **Automated Model Setup** section is present in the Solid Mechanics, Shell, and Beam interfaces. In a model that contains several physics interfaces, you should use the

automated model setup from only one of them, and make sure that all the involved interfaces are selected in the **Interfaces** list.

ADVANCED SETTINGS

You can choose how extra ODE variables added by some features are grouped in the **Dependent Variables** node of a generated solver sequence.

- Select the **Rigid materials** checkbox to group variables added by Rigid Material nodes.
- Select the **Rigid connectors** checkbox to group variables added by Rigid Connector nodes.

The selection made in the **Advanced Settings** section can be overridden by the settings in the **Advanced** section of the Rigid Material or Rigid Connector features.

DISCRETIZATION

The discretization cannot be changed. The element has different shape functions for the axial and transversal degrees of freedom. The axial displacement and twist are represented by linear shape functions, while the bending is represented by a cubic shape function ("Hermitian element").

DEPENDENT VARIABLES

The Beam interface has these dependent variables (fields):

- The displacement field $\mathbf{u}$, which has two components (`u`, `v`) in 2D and three components (`u`, `v`, and `w`) in 3D.
- The rotation angle θ, which has one component in 2D (`th`) and three components in 3D (`thx`, `thy`, and `thz`).

The names can be changed but the names of fields and dependent variables must be unique within a model.

The dependent variable names remain same in both a geometrically linear and a geometrically nonlinear analysis. Under geometric nonlinearity, the dependent variables are however not defined though shape functions. The equivalent shape function variables are (`beam.uLinx`, `beam.uLiny`, `beam.uLinz`) and (`beam.thLinx`, `beam.thLiny`, `beam.thLinz`). In this case, you will see the latter names under **Dependent Variables** in the **Solver Configurations** tree.

If needed, these shape variables can be used to write any extra contributions in the Beam interface.

If a physics interface that separates the material and spatial frame (Solid Mechanics is one such example) is added to the model, the coordinate indices change from (x, y, z) to (X, Y, Z) in the name of these variables.

- Boundary, Edge, Point, and Pair Nodes for the Beam Interface
- Modeling with Beam Elements
- Theory for the Beam Interface

- *Channel Beam*: Application Library path **Structural_Mechanics_Module/Verification_Examples/channel_beam**
- *Instability of a Space Arc Frame*: Application Library path **Structural_Mechanics_Module/Verification_Examples/space_frame_instability**

Boundary, Edge, Point, and Pair Nodes for the Beam Interface

The Beam Interface has these boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Beam interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Edge** level (3D) or **Boundary** level (2D) are

- **Material Models**
- **Line and Volume Loads**
- **Mass, Spring, and Damper**
- **Line Constraints**

The submenus at the **Point** level are

- **Connections**
- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Antisymmetry
- Attachment
- Beam End Release
- Cross-Section Data
- Edge Group
- Edge Load
- External Stress
- Hygroscopic Swelling
- Initial Stress and Strain
- Initial Values
- Linear Elastic Material
- No Rotation
- Phase
- Pinned
- Point Load
- Point Load, Free
- Point Mass
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Prescribed Acceleration
- Rigid Connector
- Section Orientation
- Section Stiffness
- Symmetry
- Thermal Expansion (for Materials)

These nodes are described for the Solid Mechanics interface:

- Added Mass
- Base Excitation
- Bolt Pretension
- Bolt Selection
- Center of Rotation Nodes
- Damping
- Fixed Constraint
- Free
- Gravity
- Inertia Relief
- Linearly Accelerated Frame[1]
- Predeformation
- Rigid Material
- Rotating Frame[1]
- Safety
- Spring Foundation
- Spring–Damper

[1] This is selected from the **Line and Volume Loads** submenu for this interface.

If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence. The exception is that the **Pinned** and **No Rotation** boundary conditions do not override each other since the degrees of freedom that they constrain are mutually exclusive.

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Initial Values

The **Initial Values** node is used to set values for the displacement field, the velocity field, the rotations, and the angular velocity. It serves as initial conditions for a transient simulation or as an initial guess for a nonlinear analysis. In addition to the default **Initial Values** node always present in the interface, you can add more **Initial Values** nodes if needed.

INITIAL VALUES

Enter values or expressions for the following based on space dimension:

- **Displacement field u**

- **Velocity field** $\frac{d\mathbf{u}}{dt}$
- **Rotation field** θ
- **Angular velocity** $\frac{d\theta}{dt}$

Context Menus

Beam > **Initial Values**

Pipe Mechanics > **Initial Values**

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Initial Values** (3D)

Boundaries > **Initial Values** (2D)

Cross-Section Data

In the **Cross-Section Data** node you specify the geometric properties of the beam's cross section. In addition, some stress evaluation properties can be defined.

For 3D models, a default Section Orientation subnode is added, in which you specify the orientation of the principal axes of the section. You can add any number of **Section Orientation** subnodes if the same section appears with different spatial orientations in the structure.

	This is required input data.

	• Common Cross Sections • Beam Cross Section

CROSS-SECTION DEFINITION

Select a **Section type** — **Rectangle**, **Box**, **Circular**, **Pipe**, **H-profile**, **U-profile**, **T-profile**, **C-profile**, **Hat**, or **User defined**. Then go to the relevant section below to continue

defining the section. Each **Section type** in the table below also has a figure showing the section and its defining dimensions.

For equations and a figure see:

- Rectangular Section
- Box Section
- Circular Section
- Pipe Section
- H-Profile Section
- U-Profile Section
- T-Profile Section
- C-Profile Section
- Hat Section

Examples of how to work with cross sections: Application Library path **Structural_Mechanics_Module/Verification_Examples/channel_beam** and **Structural_Mechanics_Module/Beams_and_Shells/pratt_truss_bridge**

Rectangle

Enter values or expressions for the following.

- **Width in local y direction** h_y
- **Width in local z direction** h_z

Box

Enter values or expressions for the following.

- **Width in local y direction** h_y
- **Width in local z direction** h_z
- **Wall thickness in local y direction** t_y
- **Wall thickness in local z direction** t_z

Circular

Enter a value or expression for the **Diameter** d_o.

Pipe

Enter values or expressions for the following.

- **Outer diameter** d_o
- **Inner diameter** d_i

H-profile, U-profile, or T-profile

Enter values or expressions for the following.

- **Section height** h_y
- **Flange width** h_z
- **Flange thickness** t_y
- **Web thickness** t_z

C-profile

Enter values or expressions for the following.

- **Flange width** h_y
- **Flange thickness** t_z
- **Web height** h_z
- **Web thickness** t_y
- **Lip width** w_l
- **Lip thickness** t_l

Hat

Enter values or expressions for the following.

- **Crown width** h_v
- **Crown thickness** t_c
- **Web height** h_w
- **Web thickness** t_w
- **Flange width** w_f
- **Flange thickness** t_f

User Defined

The following table lists the basic section properties (some apply in 3D only). The default values correspond to a circular cross section with a diameter of 0.1 m. Enter values or expressions for the section properties.

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Area of cross section	A	m^2
2D and 3D	Moment of inertia about local z-axis	I_{zz}	m^4
3D only	Distance to shear center in local z direction	e_z	m
3D only	Moment of inertia about local y-axis	I_{yy}	m^4
3D only	Distance to shear center in local y direction	e_y	m
3D only	Torsional constant	J	m^4

For 3D models, the orientation of the cross section is given in Section Orientation. If the beam's cross section is a square or circle (solid or tube), the area moments of inertia are the same independent of direction, so the beam is totally symmetric and the orientation of the principal axes of the cross section is not a problem unless you are interested in looking at results defined using the local coordinate system. Such results are bending moments, shear forces, local displacements and rotations.

STRESS EVALUATION PROPERTIES

This section is only available if **User defined** is selected as the **Section type**.

Select the **Bending stress evaluation points** — **From section heights** (the default) or **From specified points**.

Stress evaluation using only section heights is meaningful only when the cross section is symmetric.

The max shear stress factor determines the ratio between the peak and the average shear stress over the cross section as described by Equation 3-264 and Equation 3-265.

From Section Heights

For **From section heights** enter values in each field for the following parameters as needed for the space dimension:

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Section height in local y direction	h_y	m
2D and 3D	Max shear stress factor in local y direction	μ_y	l
3D only	Section height in local z direction	h_z	m
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	l

From Specified Points

For **From specified points** enter values in the **Evaluation points in local system** table as needed for the space dimension. Then enter the following parameters in the applicable fields.

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Max shear stress factor in local y direction	μ_y	l
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	l

SHEAR FLEXIBILITY

This section is only available if **User defined** is selected as the **Section type**, and the **Beam formulation** is **Timoshenko**.

Enter values or expressions for the following.

- **Shear correction factor along local y-axis** κ_y (for 2D and 3D)
- **Shear correction factor along local z-axis** κ_z (for 3D only)

LOCATION IN USER INTERFACE

Context Menus

Beam > **Cross-Section Data**

Ribbon

Physics tab with **Beam** selected:

Edges > **Beam** > **Cross-Section Data**

Section Orientation

Use the **Section Orientation** subnode to define the orientation of a beam cross section using a reference point or an orientation vector. There is always one **Section Orientation** subnode for each cross section, and as many **Section Orientation** subnodes as needed can be added if the same section appears with different spatial orientations in the structure.

This node is available for 3D components.

ORIENTATION METHOD

Select the **Reference point** (the default) or **Orientation vector**. For **Reference point** enter a **Reference point defining local y direction** P.

The coordinate system is defined as follows:

The local x direction is in the edge direction. The positive edge direction can be checked by vector plotting the local edge tangent direction. The coordinates of the reference point define the local xy-plane together with the beam axis. The local coordinate system ($\mathbf{e}_{xl}$, $\mathbf{e}_{yl}$, $\mathbf{e}_{zl}$) is formed using the following algorithm:

$$\mathbf{v}_{zl} = \mathbf{e}_{xl} \times (\mathbf{p} - \mathbf{m})$$

$$\mathbf{e}_{zl} = \frac{\mathbf{v}_{zl}}{|\mathbf{v}_{zl}|}$$

$$\mathbf{e}_{yl} = \mathbf{e}_{zl} \times \mathbf{e}_{xl}$$

Here, **p** is the reference point, and **m** is the midpoint of the beam element. The definition of the local coordinate system is illustrated in Figure 9-1.

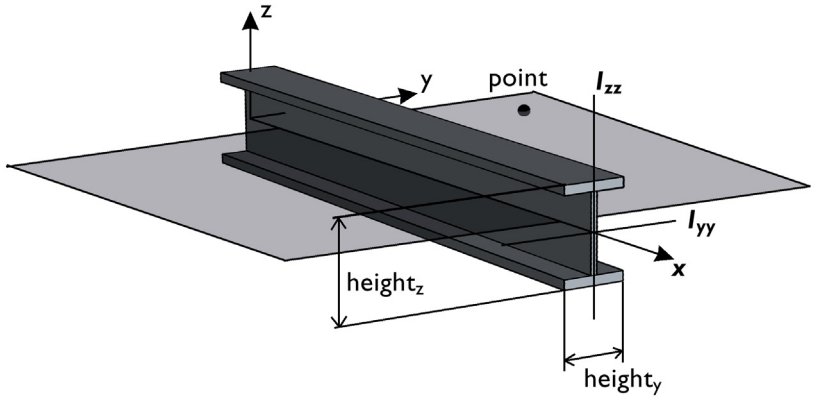


Figure 9-1: Local beam coordinate system defined by a reference point.

For the creation of a local coordinate system to be possible, the point cannot coincide with the edge or the edge extension. If this is attempted, an error message is generated.

The settings for the global coordinates of the point are `[1000,1000,1000]`. This is useful only for symmetric cross sections.

Often a number of edges in a plane have the same orientation. It is then easy to select all edges and specify a point anywhere in the same plane, not coinciding with an edge or an edge extension.

For **Orientation vector** enter **Orientation vector defining local y direction**, $\mathbf{V}$, and optionally the **Rotation of vector around beam axis** ϕ. The beam orientation is defined similarly to what is described above, with the difference that in this case the direction vector is explicitly defined whereas when an orientation point is used, the direction vector is obtained as the vector from the beam axis to the specified point. The local coordinate system ($\mathbf{e}_{xl}$, $\mathbf{e}_{yl}$, $\mathbf{e}_{zl}$) is formed using the following algorithm:

$$\mathbf{v}_{z1} = \mathbf{e}_{x1} \times \mathbf{V}$$

$$\mathbf{e}_{z1} = \frac{\mathbf{v}_{z1}}{|\mathbf{v}_{z1}|}$$

$$\mathbf{e}_{yl} = \mathbf{e}_{zl} \times \mathbf{e}_{xl}$$

The **Rotation of vector around beam axis** has the effect of rotating the given vector around the beam axis (using the right-hand rule) before it is used to define the local *xy*-plane. This simplifies the input for some cross sections, such as L-shaped profiles, where the principal axes have a direction which is skewed relative to a more natural modeling position. This can be written as

$$\mathbf{e}_{yl} = \mathbf{e}'_{yl}\cos\phi - \mathbf{e}'_{zl}\sin\phi$$

$$\mathbf{e}_{zl} = \mathbf{e}'_{zl}\cos\phi + \mathbf{e}'_{yl}\sin\phi$$

Here the directions denoted with a prime are unrotated beam axis orientations obtained by the procedure described above.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Cross-Section Data** > **Section Orientation**

Beam > **Section Stiffness** > **Section Orientation**

Ribbon

Physics tab with **Cross-Section Data** or **Section Stiffness** node selected in the model tree:

Attributes > **Section Orientation**

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic beam and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Damping
- Safety

LINEAR ELASTIC MATERIAL

Define the linear elastic material properties.

Specification of Elastic Properties for Isotropic Materials

From the **Specify** list, select a pair of elastic properties for an isotropic material. Select:

- **Young's modulus and Poisson's ratio** to specify Young's modulus (elastic modulus) E and Poisson's ratio ν. Poisson's ratio is used for computing the torsional stiffness, and is thus important only for 3D beams.
- **Young's modulus and shear modulus** to specify Young's modulus (elastic modulus) E and the shear modulus G. The shear modulus is used for computing the torsional stiffness, and is thus important only for 3D beams.
- **Bulk modulus and shear modulus** to specify the bulk modulus K and the shear modulus G.
- **Lamé parameters** to specify the Lamé parameters λ and μ.
- **Pressure-wave and shear-wave speeds** to specify the pressure-wave speed (longitudinal wave speed) c_p and the shear-wave speed (transverse wave speed) c_s.

This is the wave speed for a solid continuum. In a truss or beam element, the actual speed with which a longitudinal wave travels is lower than the value given. When using this type of input the density must also be given.

For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Each of these pairs define the elastic properties, and it is possible to convert from one set of properties to another.

Density

Define the **Density** ρ of the material. Select **From material** to take the value from the material or **User defined** to enter a value for the density.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

Thermally Loaded Beam: Application Library path
Structural_Mechanics_Module/Verification_Examples/thermally_loaded_beam

GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large rotation formulation for all edges. Select the **Geometrically linear formulation** checkbox to always use a small rotation formulation for the edges that have this material assigned, irrespective of the setting in the study step.

LOCATION IN USER INTERFACE

Context Menus
Beam > **Material Models** > **Linear Elastic Material**

Ribbon
Physics tab with **Beam** selected:

Edges > **Material Models** > **Linear Elastic Material** (3D)
Boundaries > **Material Models** > **Linear Elastic Material** (2D)

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature. The thermal strain depends on the coefficient of thermal expansion (CTE) α, the temperature T, and the strain-free reference temperature T_{ref} as

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

It is possible to model bending due to a temperature gradient in the transverse directions of the beam. The temperature is then assumed to vary linearly through the thickness.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

From the **Temperature** T list, select an existing temperature variable from a heat transfer interface. For **User defined** enter a value or expression for the temperature (the default

is 293.15 K). This is the centerline temperature of the beam, controlling the axial part of the thermal expansion.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*
- Temperature-Dependent Material Data

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain.

From the **Coefficient of thermal expansion**, α, list, select **From material** to use the coefficient of thermal expansion from the material, or **User defined** to enter a value or expression for α.

THERMAL BENDING

Enter the **Temperature gradient in local y direction** T_{gy} (in 2D and 3D) and in the **Temperature gradient in local z direction** T_{gz} (in 3D), which affects the thermal bending. If beam cross-section dimensions have been defined at **Bending stress evaluation points** — **From section heights**, these could be used in an expression containing the temperature difference.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Linear Elastic Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the transverse directions of the beam. The concentration is then assumed to vary linearly through the thickness.

MODEL INPUTS

From the **Concentration** c list, select an existing concentration variable from another physics interface, if any concentration variables exist. For **User defined** enter a value or expression for the concentration. This is the centerline concentration of the beam, controlling the axial part of the hygroscopic swelling.

The unit for the input depends on the setting of **Concentration type** in the **Hygroscopic Swelling Properties** section. Only concentration variables having the chosen physical dimension are available in the **Concentration** list.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** (the default) or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref}. This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as **Concentration type**, also enter the **Molar mass** of the fluid, M_m. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_h uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of hygroscopic swelling tensor β_h. The default value for the **User defined** case is `1.5e-4` m^3/kg.

HYGROSCOPIC BENDING

Enter the **Concentration gradient in local y direction** c_{gy} (in 2D and 3D) and in the **Concentration gradient in local z direction** c_{gz} (in 3D), which affects the hygroscopic bending. If beam cross-section dimensions have been defined at **Bending stress evaluation points** — **From section heights**, these could be used in an expression containing the concentration difference.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Linear Elastic Material** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** to a **Linear Elastic Material**, **Section Stiffness**, or **Fluid and Pipe Properties** node, in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense, but rather a contribution to the constitutive relation.

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

INITIAL STRESS

Specify the initial stress as the:

- **Initial axial force** N_{i}.
- **Initial bending moment** $M_{\mathrm{i}z}$ and for 3D models, $M_{\mathrm{i}y}$.
- *For 3D models*: **Initial torsional moment** $M_{\mathrm{i}x}$.
- *For Timoshenko beam*: **Initial shear force** $T_{\mathrm{i}y}$ and for 3D models, $T_{\mathrm{i}z}$.

INITIAL STRAIN

Specify the initial strain as the:

- **Initial axial strain** e_{ni}.
- **Initial curvature** $\theta_{\mathrm{si}z}$ and for 3D models, and $\theta_{\mathrm{si}y}$.
- *For 3D models*: **Initial twist** $\theta_{\mathrm{si}x}$.
- *For Timoshenko beam*: **Initial shear strain** $\gamma_{\mathrm{si}y}$ and for 3D models, $\gamma_{\mathrm{si}z}$.

For details about initial stresses and strains, see Inelastic Strain Contributions and Initial Stress and Strain.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Linear Elastic Material** > **Initial Stress and Strain**

Beam > **Section Stiffness** > **Initial Stress and Strain**

Pipe Mechanics > **Fluid and Pipe Properties** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material**, **Section Stiffness**, or **Fluid and Pipe Properties** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode to a **Linear Elastic Material**, **Section Stiffness**, or **Fluid and Pipe Properties** node, in order to specify an additional stress contribution which is not part of the constitutive relation. The external stress can be added to the total stress tensor, or act only as an extra load contribution.

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor.

EXTERNAL STRESS

Specify the external stress as section forces:

- **External axial force** N_{ext}.
- **External bending moment** $M_{z,\text{ext}}$ and for 3D models, $M_{y,\text{ext}}$.
- *For 3D models*: **External torsional moment** $M_{x,\text{ext}}$.
- *For Timoshenko beam*: **External shear force** $T_{y,\text{ext}}$ and for 3D models, $T_{z,\text{ext}}$.

For each quantity, you can select **User defined**, in which case you enter values or expressions explicitly. The lists will also contain any section forces announced from a physics interface.

Selecting section forces announced by the same physics interface as where the **External Stress** node is added, will result in an error ('`Circular variable dependency detected`'). This operation would imply that the computed force depends on itself.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Linear Elastic Material** > **External Stress**
Beam > **Section Stiffness** > **External Stress**
Pipe Mechanics > **Fluid and Pipe Properties** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material**, **Section Stiffness**, or **Fluid and Pipe Properties** node selected in the model tree:

Attributes > **External Stress**

Section Stiffness

The **Section Stiffness** material model provides a way of directly entering the stiffness of a beam cross section without direct knowledge of the geometry or materials used. It provides a method for entering data for complex geometries and cross sections made of multiple materials. This is done by specifying the equivalent stiffness of the cross-section, as well as several other cross-sectional properties such as mass and mass moments of inertia.

Since only the stiffness is known, but not the cross-section geometry, it is not possible to deduce any sectional stresses. You can, however, provide an expression for stress computation based on the computed section forces.

For 3D models, a default Section Orientation subnode is added, in which you specify the orientation of the principal axes of the section. You can add any number of **Section Orientation** subnodes if the same section appears with different spatial orientations in the structure.

By adding the following subnodes to the **Section Stiffness** node you can incorporate other effects:

- Initial Stress and Strain
- External Stress
- Damping

SECTION PROPERTIES

Specify the components of the **Stiffness matrix**, $\mathbf{S}$. The matrix is always assumed to be symmetric, but the number of components required vary depending on the beam formulation and the space dimension:

- For a 3D Euler–Bernoulli beam, enter the 4-by-4 matrix. The columns, in order, correspond to the axial force N, torsional moment M_{xl}, and bending moments M_{yl} and M_{zl}. The rows, in order, correspond to the axial strain ε_{n}, axial twist κ_{xl}, and bending curvatures κ_{yl} and κ_{zl}.
- For a 2D Euler–Bernoulli beam, enter the 2-by-2 matrix. The columns, in order, correspond to the axial force N and bending moments M_{zl}. The rows, in order, correspond to the axial strain ε_{n}, and curvature κ_{zl}.
- For a 3D Timoshenko beam, enter the 6-by-6 matrix. The columns, in order, correspond to the axial force N, shear forces T_{yl} and T_{zl}, torsional moment M_{xl}, and bending moments M_{yl} and M_{zl}. The rows, in order, correspond to the axial strain ε_{n}, shear strains γ_{yl} and γ_{zl}, axial twist κ_{xl}, and bending curvatures κ_{yl} and κ_{zl}.
- For a 2D Timoshenko beam, enter the 3-by-3 matrix. The columns, in order, correspond to the axial force N, shear force T_{yl}, and bending moment M_{zl}. The rows, in order, correspond to the axial strain ε_{n}, shear strain γ_{yl}, and bending curvature κ_{zl}.

In 3D, also enter the **Distance to shear in local z direction**, e_{z}, and the **Distance to shear center in local y direction**, e_{y}. These two inputs can be used to specify a distance between the center of mass and the shear center of the cross-section. If the inputs are nonzero, an applied edge load will also create a twisting moment on the cross-section.

Specify the **Mass per unit length**, m_{L}, and the **Mass moment of inertia per unit length**, $\mathbf{I}_{\mathrm{m}}$, of the beam in order to define the inertial forces. In 2D, only the **Mass moment of inertia per unit length, zz-component**, $I_{\mathrm{m,zz}}$, is needed.

In addition, specify the **First moment of mass per unit length, local y-component**, $m_{1\mathrm{y}}$, and the **First moment of mass per unit length, local z-component**, $m_{1\mathrm{z}}$, if the modeled

point does not coincide with the center of mass; otherwise keep these inputs equal to zero.

The inertia inputs are needed only for dynamic analysis. They are also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large rotation formulation for all edges. Select the **Geometrically linear formulation** checkbox to always use a small rotation formulation for the edges that have this material assigned, irrespective of the setting in the study step.

STRESS EVALUATION PROPERTIES

If you want a certain stress value to be computed, you can enter a linear relation between section forces and stress here.

Enter the **Evaluation factors in local system**. In 3D, there are four evaluation points available in the cross-section. The second column in the table corresponds to the coefficient for the axial force N, the third to the bending moment around the local y-axis M_{yl}, and the fourth to the bending moment around the local z-axis M_{zl}. In 2D, there are two evaluation points available in the cross-section. Only the coefficients for the normal force and the bending moment around the local z-axis M_{zl} are required.

In addition, in 3D enter a **Twisting moment factor**, S_T, a **Shear force factor, local y direction**, S_{Sy}, and a **Shear force factor, local z direction**, S_{Sz}. In 2D, only the **Shear force factor, local y direction** is needed.

By default, the table and the additional stress evaluation factors are populated with coefficients that correspond to a fictitious rectangular cross-section made of a single homogeneous material. The evaluation points then correspond to the four corners of the cross-section.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Material Models** > **Section Stiffness**

Ribbon

Physics tab with **Beam** selected:

Edges > **Material Models** > **Section Stiffness** (3D)
Boundaries > **Material Models** > **Section Stiffness** (2D)

Prescribed Displacement/Rotation

The **Prescribed Displacement/Rotation** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition where the displacements and rotations are prescribed in one or more directions. It is also possible set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

- If a prescribed displacement is not activated in any direction, this is the same as a **Free** constraint.
- If zero displacements and rotations are prescribed, this is the same as a **Fixed Constraint**.
- If only zero displacements are prescribed, this is the same as a **Pinned** constraint.
- If only zero rotations are prescribed, this is the same as a **No rotation** constraint.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the prescribed displacement/rotation. See the section **Coordinate System Selection** for Edge Load.

PRESCRIBED DISPLACEMENT

For the displacement in each direction, select a setting from the list — **Free**, **Prescribed**, or **Limited**. Select:

- **Free** (the default) to leave the displacement component unconstrained
- **Prescribed** to constrain the displacement component to a given value. Enter a scalar value for the component of the prescribed displacement u_0.
- **Limited** to set a maximum and a minimum limit for the displacement component. Enter a scalar value for the component of the maximum displacement $u_{0,\max}$ and the minimum displacement $u_{0,\min}$. By default, they are set to `Inf` and `-Inf`, which corresponds to no active constraint.

If any displacement component is set to **Limited**, an additional section **Limited displacement** is visible. Select the **Method** used to implement the weak inequality

constraint — **Penalty** or **Augmented Lagrangian**. For both methods, enter a **Penalty factor** k_p.

By default, the **Penalty** method is suggested, which in principle enforces the maximum and minimum limits for the displacement by adding nonlinear springs with a stiffness equal to k_p when the limits are exceeded. This method is usually robust, but the accuracy is directly dependent on the chosen penalty factor.

The **Augmented Lagrangian** method adds extra degrees of freedom to improve the accuracy of the constraint. Here, the penalty factor is a numerical parameter, and has less impact on the accuracy of the constraint compared to when using the penalty method. The implementation of the augmented Lagrangian method puts no restrictions on the solver sequence, but for good convergence, proper scaling of the extra degrees of freedom can be important.

The default value for the **Penalty factor** k_p depends on what type of entity the **Prescribed Displacement/Rotation** node is added to.

- For points, the default expression is
  `100*beam.Eequ*<phys>.area*<phys>.re^2/<phys>.<pdr>.charLen^3`
- For edges, the default expression is
  `100*<phys>.Eequ*<phys>.area*<phys>.re^2/<phys>.<pdr>.charLen^4`

In these expressions, `<phys>` is the tag of the physics interface (for example `beam` or `pipem`) and `<pdr>` is the tag of the **Prescribed Displacement/Rotation** node. The expressions are given as an estimation to the bending stiffness, where `<phys>.Eequ` is the equivalent Young's modulus, `<phys>.re` is the equivalent radius of gyration, and `<phys>.area` is the cross-section area. The variable `<phys>.<tag>.charLen` is by default equal to the length of the mesh element. To improve the estimate of the penalty factor, replace `<phys>.<tag>.charLen` with the free length of the beam or pipe. The penalty factor can also be tuned by changing the multiplier at the beginning of each expression.

For details about maximum and minimum limits for the displacements, see Limited Displacement.

PRESCRIBED ROTATION

For 2D models, to define a prescribed rotation select the **Prescribed in out of plane direction** checkbox and enter a value or expression for θ_{0z}.

For 3D models, select a prescribed rotation from the list — **Free** or **Rotation**. Select:

- **Free** (the default) to leave the rotations unconstrained.
- **Rotation** to activate a prescribed rotation. Enter values or expressions for the prescribed rotation vector θ. When the study is geometrically linear, you directly prescribe the individual components of the rotation vector. Under **For geometric linearity**, select one or several of the **Free rotation around x direction**, **Free rotation around y direction**, and **Free rotation around z direction** checkboxes to remove the constraint for the corresponding rotation component. If cleared, the rotations are constrained to either the input value or to the default zero rotation. The status of the checkboxes has no effect when geometric nonlinearity is activated under the study settings. This is because the constraints put on different rotation components are not independent of each other in the case of finite rotations. Here, θ should be interpreted as a rotation vector in the given coordinate system. The norm of the vector is the angle of rotation, and the orientation of the axis of rotation is given by the vector components.

In a geometrically nonlinear analysis in 3D, you should prescribe all three components of the rotation vector. Prescribing only one or two components may not give unique results, since finite rotations are not commutative.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements in a frequency domain analysis of perturbation type.
- You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.
- You can assign the value of the prescribed displacement to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Prescribed Displacement**
Beam > **Prescribed Displacement** (Point)
Pipe Mechanics > **Line Constraints** > **Prescribed Displacement**
Pipe Mechanics > **Prescribed Displacement** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Prescribed Displacement** (3D)
Boundaries > **Line Constraints** > **Prescribed Displacement** (2D)
Points > **Beam** > **Prescribed Displacement**

Prescribed Velocity

The **Prescribed Velocity** node adds an edge (3D), boundary (2D), or point (2D and 3D) that prescribes the translational or rotational velocity in one or more directions.

The prescribed velocity condition is applicable for Time Dependent and Frequency Domain studies. With this condition it is possible to prescribe a velocity in one direction, leaving the beam free in the other directions.

The Prescribed Velocity node is a constraint, and overrides any other constraint on the same selection.

	For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

COORDINATE SYSTEM SELECTION

	Coordinate systems with directions that change with time should not be used.

Specify the coordinate system to use for specifying the prescribed translational/rotational velocity. See the section **Coordinate System Selection** for Edge Load. Depending on the selected coordinate system, the velocity components change accordingly.

PRESCRIBED VELOCITY

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and for 3D models, **Prescribed in z direction** checkboxes. Then enter a value or expression for v_x, v_y, and for 3D models, v_z.

PRESCRIBED ANGULAR VELOCITY

	For 2D models, to define a prescribed angular velocity select the **Prescribed in out of plane direction** checkbox and enter a value or expression for $\partial\theta_{0z}/\partial t$.

	For 3D models, to define a prescribed angular velocity for each space direction (x, y, and z), select one or all of the **Prescribed around x direction**, **Prescribed around y direction**, and **Prescribed around z direction** checkboxes and enter a value or expression for in each $\partial\theta_{0x}/\partial t$, $\partial\theta_{0y}/\partial t$, or $\partial\theta_{0z}/\partial t$ field.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed velocity in a frequency domain analysis of perturbation type.
- **Prescribed Velocity** cannot be used as a weak constraint.

Location in User Interface

Context Menus

Beam > **Line Constraints** > **Prescribed Velocity**
Beam > **More Constraints** > **Prescribed Velocity** (Point)
Pipe Mechanics > **Line Constraints** > **Prescribed Velocity**
Pipe Mechanics > **More Constraints** > **Prescribed Velocity** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Prescribed Velocity** (3D)
Boundaries > **Line Constraints** > **Prescribed Velocity** (2D)
Points > **More Constraints** > **Prescribed Velocity**

Prescribed Acceleration

The **Prescribed Acceleration** node adds a boundary or domain condition where the acceleration is prescribed in one or more directions. The prescribed acceleration condition is applicable for Time Dependent and Frequency Domain studies. With this boundary condition it is possible to prescribe an acceleration in one direction, leaving the beam free in the other directions.

The Prescribed Acceleration node is a constraint, and overrides any other constraint on the same selection.

For details about prescribed velocities and accelerations, see Prescribed Displacements, Velocities, and Accelerations.

COORDINATE SYSTEM SELECTION

Coordinate systems with directions that change with time should not be used.

Specify the coordinate system to use for specifying the prescribed translational/rotational acceleration. See the section **Coordinate System Selection** for Edge Load. Depending on the selected coordinate system, the acceleration components change accordingly.

PRESCRIBED ACCELERATION

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and for 3D models, **Prescribed in z direction** checkboxes. Then enter a value or expression for a_x, a_y, and for 3D models, a_z.

PRESCRIBED ANGULAR ACCELERATION

	For 2D models, to define a prescribed angular acceleration select the **Prescribed in out of plane direction** checkbox and enter a value or expression for $\partial^2\theta_{0z}/\partial t^2$.
	For 3D models, to define a prescribed angular acceleration for each space direction (x, y, and z), select one or all of the **Prescribed around x direction**, **Prescribed around y direction**, and **Prescribed around z direction** checkboxes and enter a value or expression for in each $^2\theta_{0x}/\partial t^2$, $^2\theta_{0y}/\partial t^2$, or $^2\theta_{0z}/\partial t^2$ field.
	• You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed acceleration in a frequency domain analysis of perturbation type. • **Prescribed Velocity** cannot be used as a weak constraint.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Prescribed Acceleration**

Beam > **More Constraints** > **Prescribed Acceleration** (Point)

Ribbon

Physics tab with **Beam** selected:

Edges > **Line Constraints** > **Prescribed Acceleration**

Points > **More Constraints** > **Prescribed Acceleration**

Context Menus

Beam > **Line Constraints** > **Prescribed Acceleration**
Beam > **More Constraints** > **Prescribed Acceleration** (Point)
Pipe Mechanics > **Line Constraints** > **Prescribed Acceleration**
Pipe Mechanics > **More Constraints** > **Prescribed Acceleration** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Prescribed Acceleration** (3D)
Boundaries > **Line Constraints** > **Prescribed Acceleration** (2D)
Points > **More Constraints** > **Prescribed Acceleration**

Pinned

The **Pinned** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that makes all nodes on the selected objects to have zero displacements; that is, all translations are fixed while still allowing rotations.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Pinned** (Line)
Beam > **Pinned** (Point)
Pipe Mechanics > **Line Constraints** > **Pinned** (Line)

Pipe Mechanics > **Pinned** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Pinned** (3D)
Boundaries > **Line Constraints** > **Pinned** (2D)
Points > **Beam** > **Pinned**

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Fixed Constraint**, **Prescribed Displacement**, **Pinned**, **No Rotation**, **Rigid Connector**, or **Attachment**) to prescribe a deformation of the constraint caused by changes in temperature of the surroundings. This makes it possible to reduce stresses caused by the boundary conditions.

The thermal strain depends on the coefficient of thermal expansion α, the temperature T, and the strain-free reference temperature T_{ref} as

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from edge** to take the thermal expansion data from the edge being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the edge itself.

When **Inherit from edge** is not selected, enter:

- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** α. As a default, values **From material** are used. This requires that a material has been assigned to the points where the constraint is

active.

For **User defined** enter the coefficient of thermal expansion α.

- A value or expression for the **Volume reference temperature** T_{ref} which is the temperature at which there are no thermal displacements at the constraints.

Enter the coordinates of the **Reference point**, which is the point where the displacement and rotation are zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Fixed** > **Thermal Expansion**
Beam > **Prescribed Displacement** > **Thermal Expansion**
Beam > **Pinned** > **Thermal Expansion**
Beam > **No Rotation** > **Thermal Expansion**
Beam > **Rigid Connector** > **Thermal Expansion**
Beam > **Attachment** > **Thermal Expansion**
Pipe Mechanics > **Fixed** > **Thermal Expansion**
Pipe Mechanics > **Prescribed Displacement** > **Thermal Expansion**
Pipe Mechanics > **Pinned** > **Thermal Expansion**
Pipe Mechanics > **No Rotation** > **Thermal Expansion**
Pipe Mechanics > **Rigid Connector** > **Thermal Expansion**

Ribbon

Physics tab with **Fixed**, **Prescribed Displacement**, **Pinned**, **No Rotation**, **Rigid Connector**, or **Attachment** node selected in the model tree:

Attributes > **Thermal Expansion**

No Rotation

The **No Rotation** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that prevents all rotation at the selected objects while still allowing translational motion.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **No Rotation** (Line)
Beam > **More Constraints** > **No Rotation** (Point)
Pipe Mechanics > **Line Constraints** > **No Rotation** (Line)
Pipe Mechanics > **More Constraints** > **No Rotation** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **No Rotation** (3D)
Boundaries > **Line Constraints** > **No Rotation** (2D)
Points > **More Constraints** > **No Rotation**

Symmetry

The **Symmetry** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that defines asymmetry edge, boundary, or point.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the orientation of the symmetry plane. See the section **Coordinate System Selection** for Edge Load.

SYMMETRY

Select an **Axis to use as symmetry plane normal** to specify the direction of the normal to the symmetry plane in the selected coordinate system.

- For 2D models, select **1** or **2** for the first or second axis, respectively.
- For 3D models, select **1**, **2**, or **3** for the first, second, or third axis, respectively.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_{n}. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.beam.sym1.un`.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Symmetry**

Beam > **More Constraints** > **Symmetry** (Point)

Pipe Mechanics > **Line Constraints** > **Symmetry**

Pipe Mechanics > **More Constraints** > **Symmetry** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Symmetry** (3D)
Edges > **Line Constraints** > **Symmetry** (2D)
Points > **More Constraints** > **Symmetry**

Antisymmetry

The **Antisymmetry** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that defines an antisymmetry edge, boundary, or point.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the antisymmetry. See the section **Coordinate System Selection** for Edge Load.

ANTISYMMETRY

Select an **Axis to use as anti-symmetry plane normal** to specify the direction of the normal to the symmetry plane in the selected coordinate system.

- For 2D models, select **1** or **2** for the first or second axis, respectively.
- For 3D models, select **1**, **2**, or **3** for the first, second, or third axis, respectively.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Antisymmetry**
Beam > **More Constraints** > **Antisymmetry** (Point)
Pipe Mechanics > **Line Constraints** > **Antisymmetry**
Pipe Mechanics > **More Constraints** > **Antisymmetry** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line Constraints** > **Antisymmetry** (3D)
Boundaries > **Line Constraints** > **Antisymmetry** (2D)
Points > **More Constraints** > **Antisymmetry**

Edge Load

Add an **Edge Load** as a force or moment distributed along an edge.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the load. From the **Coordinate system** list select from the following based on space dimension:

For 2D models (boundaries):

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate system

	For 3D models (edges): • **Global coordinate system** (the default; the standard global coordinate system). • **Local edge system**. This is the coordinate system defined by the beam cross-section orientation as defined in the **Section Orientation** subnode under **Cross Section Data**. The local edge coordinate systems directions are available as variables for plotting using an Arrow Line plot, for example: Under **Beam>Beam Local System** in the plot settings lists of predefined expressions, select **Base vector (beamsys) x**, **Base vector (beamsys) y**, or **Base vector (beamsys) z**. • Any additional user-defined coordinate system.

FORCE

Select a **Load type** — **Force per reference volume**, **Force per deformed volume**, **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for the components.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference volume	$\mathbf{f}_V$	N/m	edges	3D (x, y, z)
			boundaries	2D (x, y)
Force per deformed volume	$\mathbf{f}_v$	N/m^3	edges	3D (x, y, z)
			boundaries	2D (x, y)
Force per reference length	$\mathbf{f}_L$	N/m	edges	3D (x, y, z)
			boundaries	2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m^3	edges	3D (x, y, z)
			boundaries	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	edges	3D (x, y, z)
			boundaries	2D (x, y)

- When **Force per reference volume** or **Force per deformed volume** is selected, the given load is multiplied by the cross-section area. This option is useful for modeling body loads like gravity or centrifugal loads.
- In the Beam interface, the geometrically nonlinear formulation only accounts for finite rotations, but not for finite strains. This means that the options **Force per deformed volume** and **Force per deformed length** will only affect the orientation of the load. There is no effect of centerline extension.
- After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the Beam interface with another physics interface, it is also possible to choose a predefined load from this list.

MOMENT

Enter values or expressions for the components of the **Moment per reference length** $\mathbf{m}_L$ or **Moment per deformed length** $\mathbf{m}_l$.

In the Beam interface, the geometrically nonlinear formulation only accounts for finite rotations, but not for finite strains. This means that the option **Moment per deformed length** will only affect the orientation of the load. There is no effect of centerline extension.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line and Volume Loads** > **Edge Load**
Pipe Mechanics > **Line and Volume Loads** > **Edge Load**
Beam Rotor > **Edge Load**

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Edges > **Line and Volume Loads** > **Edge Load** (3D)
Boundaries > **Line and Volume Loads** > **Edge Load** (2D)

Physics tab with **Beam Rotor** selected:

Edges > **Edge Load**

Point Load

Add a **Point Load** to points for concentrated forces or moments at points. The loads are defined in the given coordinate system.

FORCE

Select the **Load type** — **Force per point** or **Total force**. Then enter values or expressions based on the selection of the point load $\mathbf{F}_P$ or total force $\mathbf{F}_{tot}$.

The **Load** list normally only contains **User defined**. When combining the Beam interface with another physics interface, it is also possible to choose a predefined load from this list.

MOMENT

Select the **Moment type** — **Moment per point** or **Total moment**. Then enter values or expressions based on the selection of the point moment $\mathbf{M}_P$ or total moment $\mathbf{M}_{tot}$. The given moment is applied to each of the selected points.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Point Load**

Pipe Mechanics > **Point Load**

Beam Rotor > **Point Load**

Ribbon

Physics tab with **Beam**, **Pipe Mechanics**, or **Beam Rotor** selected:

Points > **Beam** > **Point Load**

Point Load, Free

Add a **Point Load, Free** node to describe concentrated loads with locations specified by coordinates. Such loads do not have to be placed in a geometrical point or in a node point. This is particularly useful for the following cases:

- Moving loads
- Several point loads, in which case it may be impractical to create points at all load locations

- For each free point load, there will be a search for the mesh element that is closest to the given location. For the load to be active, the location must be within a certain small distance from at least one element. The actual load location is taken as the point on the element that is closest to the given location. If no element is found, the load is silently ignored.

  You can check the number of loads that were ignored through the variable `<phys>.<load_tag>.num_ignored`, for example `beam.plf1.num_ignored`.
- The local stress field in the element where a point load acts will not be reliable.

LOCATION AND FORCE

For each row in the table, enter the data for one load. If material frame input is used, then the location is specified in terms of the material coordinates (**X**, **Y**, **Z**). If spatial frame input is used, then the spatial coordinates (**x**, **y**, **z**) are used. Then, enter the force and moment values, **Fxl**, **Fyl**, **Fzl**, **Mxl**, **Myl**, and **Mzl**. The force and moment vectors are interpreted in the coordinate system selected in the **Coordinate System Selection** section.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

Pratt Truss Bridge: Application Library path
Structural_Mechanics_Module/Beams_and_Shells/pratt_truss_bridge

Context Menus
Beam > **Point Load, Free**
Pipe Mechanics > **Point Load, Free**

Ribbon
Physics tab with **Beam** selected:

Global > **Beam** > **Point Load, Free**

Physics tab with **Pipe Mechanics** selected:

Global > **Pipe Mechanics** > **Point Load, Free**

Point Mass

Use the **Point Mass** node to model a discrete mass or mass moment of inertia which is concentrated at a point.

COORDINATE SYSTEM SELECTION

With the **Coordinate system** list, select the coordinate system in which the mass moment of inertia tensor is defined.

POINT MASS

Enter a **Point mass** m.

Enter a value or values for the **Mass moment of inertia**. For 2D models, enter one value for J_z. For 3D models, enter a single value for an isotropic moment of inertia tensor, or select **Diagonal** or **Symmetric** to enter a full moment of inertia tensor.

DAMPING

Enter a **Mass damping parameter** α_{dM}. This is the mass proportional term of a Rayleigh damping.

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Mass, Spring, and Damper** > **Point Mass**

Pipe Mechanics > **Mass, Spring, and Damper** > **Point Mass**

Ribbon

Physics tab with **Beam** or **Pipe Mechanics**, selected:

Points > **Mass, Spring, and Damper** > **Point Mass**

Rigid Connector

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. The selected points will move as a single rigid object.

You can add the **Rigid Connector** node at the edge (2D: boundary) and point levels.

- When added at the edge level you can connect edges and points as long as at least one edge is selected. Selecting points is optional.
- When added at the point level, you can connect a set of points to each other.

If the study step is geometrically nonlinear, the rigid connector takes finite rotations into account.

Rigid connectors are available in the Solid Mechanics, Multibody Dynamics, Shell, Beam, and Pipe Mechanics interfaces. Rigid connectors from different interfaces can be attached to each other.

You can add functionality to the rigid connector through the following subnodes:

- Applied Force (Rigid Connector) to apply a force in given point.
- Applied Moment (Rigid Connector) to apply a moment.
- Mass and Moment of Inertia (Rigid Connector) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Connector) to add a translational or rotational spring or damper in a given point.

When physics symbols are shown, a rigid connector is represented by a symbol at the center of rotation, together with a set of lines connecting the center of rotation with the centers of gravity of the selected entities.

- When the rigid connector is added at the edge level, such symbols are shown only for the selected edges, but not for auxiliary selections of points.
- Because of the way physics symbols are evaluated, as a lightweight operation when moving between physics nodes in the model builder tree, it is sometimes not possible to determine the center of rotation. In particular, if an offset is supplied, it will not be taken into account.

EDGE SELECTION

This section is present when the **Rigid Connector** node has been added at the edge level. Select one or more edges to be part of the rigid region.

BOUNDARY SELECTION

This section is present when the **Rigid Connector** node has been added at the boundary level. Select one or more edges to be part of the rigid region.

POINT SELECTION

This section is always present.

- When the **Rigid Connector** is added at the point level, select a number of points that form the rigid region.
- When the **Rigid Connector** is added at the edge (2D: boundary) level, this section is initially collapsed. Here, you can add optional points to the rigid region. The points cannot be adjacent to the selected boundaries or edges.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed displacements or rotations are specified along the axes of this coordinate system. It is also used for defining the axis directions of the moment of inertia tensor of the **Mass and Moment of Inertia** subnode.

In order to be applicable, a coordinate system must have axis directions that are independent of the location in space.

CENTER OF ROTATION

The center of rotation serves two purposes.

- If you prescribe the displacement of the rigid connector, this is the place where it is fixed.
- Results are interpreted with respect to the center of rotation.

Select a **Center of rotation** — **Automatic**, **Centroid of selected entities**, or **User defined**.

- For **Automatic** the center of rotation is at the geometrical center of the selected points. The constraints are applied at the center of rotation.

- For **Centroid of selected entities** a subnode for selection of the points is added to the Model Builder.
- For **User defined**, in the **Global coordinates of center of rotation** X_C table enter coordinates based on space dimension.

- Once **Centroid of selected entities** is chosen, a default Center of Rotation: Point subnode is added.
- The center of rotation is located at the centroid of the selected points, which do not need to be related to the objects to which the rigid connector is attached. The points do not even have to belong to the physics interface. As a special case, you can select a single point, and thus locate the center of rotation at a certain point.

Select the **Offset** checkbox to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\text{C}} = \mathbf{X}_{\text{C,input}} + \mathbf{X}_{\text{offset}}$$

PRESCRIBED DISPLACEMENT AT CENTER OF ROTATION

To define a prescribed displacement at the center of rotation for each space direction, select one or several of the available checkboxes then enter values or expressions for the prescribed displacements. The direction coordinate names can vary depending on the selected coordinate system.

- **Prescribed in x direction** $u_{0\text{x}}$
- **Prescribed in y direction** $u_{0\text{y}}$
- *For 3D components:* **Prescribed in z direction** $u_{0\text{z}}$

PRESCRIBED ROTATION

Specify the rotation at the center of rotation. Select from the **By** list: **Free** (the default), **Constrained rotation**, or **Prescribed rotation**.

For 2D components, the **Constrained rotation** and **Prescribed rotation** is always about the z-axis, so no component selection is necessary.

Constrained Rotation (3D Components)

For **Constrained rotation** select one or more of the available checkboxes to enforce zero rotation about the corresponding axis in the selected coordinate system:

- **Constrain rotation about x-axis**
- **Constrain rotation about y-axis**
- **Constrain rotation about z-axis**

Prescribed Rotation

- For **Prescribed rotation** enter an **Angle of rotation** ϕ_0. For 3D components also enter an **Axis of rotation** Ω for the **x**, **y**, and **z** coordinates.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can activate and deactivate the rigid connector by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.
- You can assign the value of the prescribed displacement and rotation to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

- Rigid Connector
- Harmonic Perturbation
- Load Cases

RELEASED DEGREES OF FREEDOM

In some cases it can be useful to not constrain the displacement in a certain direction. To do so, select a local **Coordinate system** for specifying the directions in which the degree of freedom will be released. The **Coordinate system** list contains only applicable coordinate systems in the model.

Select one or more of the available checkboxes to release the displacement in the corresponding axis in the selected coordinate system:

- **Release displacement in x1 direction**
- **Release displacement in x2 direction**

- *For 3D components:* **Release displacement in x3 direction**
- *For 3D components:* **Release rotation around x1 direction**
- *For 3D components:* **Release rotation around x2 direction**
- **Release rotation around x3 direction**

Note that the **Rigid Connector** introduces its own global displacement degrees of freedom (DOFs) and global rotation DOFs. When certain displacement field components are released, this reduces the number of equations used to solve for the rigid connector's global DOFs, while preserving the number of said DOFs. Therefore, it may become necessary to constrain some of the Rigid Connector's global DOFs to achieve static determinacy.

The section **Released Degrees of Freedom** is only shown if the checkbox **Use weak constraints for rigid-flexible connection** in the section **Constrain Settings** is not enabled.

REACTION FORCE SETTINGS

Select **Evaluate reaction forces** to compute the reaction force caused by a prescribed motion. The default is to not compute the reaction force. When selected, the prescribed motion is implemented as a weak constraint.

Select **Apply reaction only on rigid body variables** to use a unidirectional constraint for enforcing a prescribed motion. The default is that bidirectional constraints are used. This setting is useful in a situation where a bidirectional constraint would give an unwanted coupling in the equations. This would happen if the prescribed value of the motion is a variable solved for in other equations.

FORMULATION

Some contributions from a rigid connector will, under geometric nonlinearity, result in a nonsymmetric local stiffness matrix. If all other aspects of the model are such that the global stiffness matrix would be symmetric, then such a nonsymmetric contribution may have a heavy impact on the total solution time and memory usage. In such cases, it is often more efficient to use an approximative local stiffness matrix that is symmetric.

Select **Use symmetric formulation for geometric nonlinearity** to force all matrix contributions from the rigid connector and its subnodes to be symmetric.

Using an approximative stiffness matrix will in general require more iterations. However, since the computational cost per iteration will be cut at least by a factor of two if a symmetric matrix can be used, it is usually more efficient to ignore a weak nonsymmetry.

In particular, if the rotation of the rigid connector per time step or parameter increment is small, there will in general be no increase in the number of iterations at all if this option is used.

When the global stiffness matrix is nonsymmetric for other reasons, then there is nothing to be gained from symmetrizing the contribution from the rigid connector.

CONSTRAINT SETTINGS

On the boundaries where the rigid connector is coupled to a flexible material, all nodes on such a boundary are constrained to move as a rigid body. As a default these constraints are implemented as pointwise constraints. If you want to use a weak constraint formulation, select **Use weak constraints for rigid-flexible connection**.

This formulation cannot be combined with the **Flexible** formulation of the rigid connector, which in itself is a special form of weak constraint.

CONSTRAINT SETTINGS

On the points where the rigid connector is coupled to a flexible material, all nodes are constrained to move as a rigid body. As a default, these constraints are implemented as pointwise constraints. If you want to use a weak constraint formulation, select **Use weak constraints for rigid-flexible connection**.

ADVANCED

It is possible to couple rigid connectors to each other. In the **Connect to** list, you can select any rigid connector defined in the Solid Mechanics, Multibody Dynamics, or Shell interfaces as being rigidly connected to the current one.

Select **Group dependent variables in solver** — **From physics interface** (default), **Yes**, or **No**, to choose how to group in the solver sequence the dependent variables added by the **Rigid Connector** feature.

	When coupling rigid connectors to each other, you must be careful not to add conflicting settings. Typically, you should only assign constraints to one of the connectors, and it is recommended to use a common center of rotation. It is, however, possible to deviate from these recommendations and create other meaningful combinations.

	Rigid Connector

LOCATION IN USER INTERFACE

Context Menus

Beam > **Line Constraints** > **Rigid Connector** (3D: Edge, 2D: Boundary)
Beam > **Connections** > **Rigid Connector** (Point)
Pipe Mechanics > **Line Constraints** > **Rigid Connector** (3D: Edge, 2D: Boundary)
Pipe Mechanics > **Connections** > **Rigid Connector** (Point)

Ribbon

Physics tab with **Beam** or **Pipe Mechanics** selected:

Boundaries > **Line Constraints** > **Rigid Connector** (2D)
Edges > **Line Constraints** > **Rigid Connector** (3D)
Points > **Connections** > **Rigid Connector**

Attachment

The **Attachment** node is used to define a point or a set of points on a flexible beam which can be used to connect it with other components through a joint in the Multibody Dynamics interface. All the selected points behave as if they were connected by a common rigid body.

Attachments can be added to boundaries in a Multibody Dynamics or a Solid Mechanics interface, to edges in a Shell or Layered Shell interface, or to points in a

Beam interface. This makes it possible to use a joint in the Multibody Dynamics interface for connecting parts modeled in different physics interfaces.

The use of attachments is discussed in the Attachments section in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Connections** > **Attachment**

Ribbon

Physics tab with **Beam** selected:

Points > **Connections** > **Attachment**

Beam End Release

In cases where a joint between two or more beams is not perfectly rigid, you can add a **Beam End Release** node to specify the decoupled degrees of freedom.

If three or more beams meet at the same point, you may need to add an Edge Group subnode in order to specify how the beams are connected to each other. The default is that all beams meeting at a point are independent from each other in the selected degrees of freedom.

EDGE DEFINING THE LOCAL ORIENTATION

In the case of a geometrically nonlinear analysis, the orientation of the axes along which one or more degrees of freedom are decoupled may change due to finite rotations.

For cases when rotational degrees of freedom are decoupled, the rotation will however not be the same in all beams attached to a joint. Often this is not an issue since the interesting orientation is still unique, but you have a possibility to explicitly select the edge from which the axis rotations are inferred. The default option is **Point Evaluation**. In this case, the rotation is averaged between all participating edges.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

RELEASE SETTINGS

Select the degrees of freedom which are to be decoupled at the selected point. The directions are interpreted along the axes of the selected coordinate system. In a geometrically nonlinear analysis, the coordinate system only gives the orientation in the original configuration. The actual orientation in which the degrees of freedom are decoupled rotates with the beam.

	Adding loads or constraints to a degree of freedom that has been disconnected using **Beam End Release** may give unintuitive results. Essentially, the degree of freedom value at the point is defined as the average of the values in the disconnected beams. A prescribed displacement will thus act as a constraint equation, prescribing a weighted sum of the displacements or rotations in the connected beams, but not their individual values.

	Nonrigid Joints

	Stress Analysis of a Portal Crane: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/portal_crane**

LOCATION IN USER INTERFACE

Context Menus

Beam > **Beam End Release**

Ribbon

Physics tab with **Beam** selected:

Points > **Beam End Release**

Edge Group

Add one or more **Edge Group** subnodes to a Beam End Release node when you need to specify how beams are connected at a joint. All beams which are selected in an edge group will be fully connected at the joint. Each group of edges will be free to move

relative to any other edge group in the degrees of freedom selected in the parent **Beam End Release** node.

A certain edge can only be part of one edge group. A selection in a later edge group will override a selection of the same edge in a previous group.

Nonrigid Joints

LOCATION IN USER INTERFACE

Context Menus

Beam > **Beam End Release** > **Edge Group**

Ribbon

Physics tab with **Beam End Release** selected:

Attributes > **Edge Group**

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response, the physics interface splits the harmonic load into two parameters:

- The amplitude, F, which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f:

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi f t + \phi)$$

PHASE

Add the phase load $\mathbf{F}_{\text{ph}}$ for harmonic loads. Enter the phase for each component of the load in the corresponding fields.

MOMENT LOAD PHASE

Add the phase for the moment load $\mathbf{M}_{ph}$ for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Edge Load** > **Phase**
Beam > **Point Load** > **Phase**
Pipe Mechanics > **Edge Load** > **Phase**
Pipe Mechanics > **Point Load** > **Phase**

Ribbon

Physics tab with **Edge Load** or **Point Load** selected:

Attributes > **Phase**

Harmonic Perturbation

Use the **Harmonic Perturbation** subnode to specify the harmonic part of nonzero prescribed displacements or rotations. This node is used if the study step is frequency response of a perturbation type.

The settings are the same as in the parent Prescribed Displacement/Rotation, Prescribed Velocity, or Prescribed Displacement/Rotation node. Only degrees of freedom selected as prescribed in the parent node can be assigned a value.

See Harmonic Perturbation in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Beam > **Prescribed Velocity** > **Harmonic Perturbation**
Beam > **Prescribed Acceleration** > **Harmonic Perturbation**
Pipe Mechanics > **Prescribed Displacement/Rotation** > **Harmonic Perturbation**
Pipe Mechanics > **Prescribed Velocity** > **Harmonic Perturbation**
Pipe Mechanics > **Prescribed Acceleration** > **Harmonic Perturbation**

Ribbon

Physics tab with **Prescribed Displacement/Rotation**, **Prescribed Velocity**, or **Prescribed Acceleration** node selected in the model tree:

Attributes > **Harmonic Perturbation**

10

Beam Cross Section

This chapter describes the Beam Cross Section interface, which you find under the **Structural Mechanics** branch ( ) when adding a physics interface.

In this chapter:

- Using the Beam Cross Section Interface
- Theory for the Beam Cross Section Interface
- The Beam Cross Section Interface

Using the Beam Cross Section Interface

In this section:

- About Beams and Cross-Section Data
- Using the Beam Cross Section Interface

About Beams and Cross-Section Data

A *beam* is a slender structure that can be fully described by cross-section properties like area and moments of inertia. In COMSOL Multiphysics, there are two physics interfaces for analyzing beam structures, one in 2D and one in 3D. There are also two similar Pipe Mechanics interfaces, targeted at piping systems.

APPROACHES FOR OBTAINING THE CROSS-SECTION DATA

The values for the cross-section properties can be found in different ways. By:

- looking up tabulated values,
- inserting dimensions into handbook formulas,
- using the built-in common sections in The Beam Interface, and
- using The Beam Cross Section Interface.

For the common built-in cross sections available with the Beam interface (Rectangular, Box, Circular, Pipe, H-section, U-section, and T-section), using the **Cross-Section Data** settings provides you with the most efficient input alternative. See Common Cross Sections for details.

For nonstandard cross sections with nontrivial shapes, using the Beam Cross Section interface is the only realistic alternative. This physics interface also provides you with more accurate data than the other methods, since it does not rely on common engineering approximations.

The cross-section data obtained from the different approaches can, in general, differ somewhat. Two common sources for this difference is that the influence of fillets in the

geometry can be excluded, and that approximations such as assuming thin walled sections can be used.

STRESS COMPUTATION

Since a beam interface is only aware of the cross-section properties and not of the true stress distribution over the cross section, the stresses computed there can only be approximate. The approximations used by COMSOL are conservative in the sense that all stress contributions are assumed to interact in the worst possible way.

In the Beam Cross Section interface, the true stress distribution can be computed given a set of section forces. This can be important in cases where the stresses obtained in the Beam interface or the Pipe Mechanics interface exceed allowed values. Also, from a pedagogical point of view, it is an advantage to be able to display the full stress distribution.

With the Beam Cross Section interface in 2D, you can plot the stress distribution in a certain cross section. In 3D, the beam and its results can be plotted using a solid representation.

Using the Beam Cross Section Interface

When the document refers to 'beams' in the following, the information is applicable to both the Beam and Pipe Mechanics interfaces.

The Beam Cross Section interface can be used in three different ways:

- In a separate model, to compute beam cross section data that is manually transferred as input to another model where a beam analysis is performed. Possibly, the cross section data is also used outside COMSOL Multiphysics.
- In the same model as beam interface, but with the Beam Cross Section interface placed in a separate component. Data can then be transferred between the two interfaces by entering variable names in input fields.
- In the same component as the beam interface. In this case, multiphysics couplings can be used to transfer data between the interfaces.

When using the Beam Cross Section interface and the beam interface in the same model, there are a number of things to pay attention to:

- Usually the determination of the cross-section data is computationally more expensive than the actual analysis of the beam structure. Thus, it is best to use either separate studies or separate study steps for the two tasks.
- If two separate studies or study steps are used, then the **Values of variables not solved for** must be set in the second study step, where only the beam problem is solved. Under **Dependent Variables**, you can also clear the **Store in output** checkbox for the beam cross section degrees of freedom in order to save space.
- You can transfer data between the two physics interfaces using the Beam Cross Section–Beam Coupling and Beam–Beam Cross Section Coupling multiphysics couplings if the interfaces belong to the same component.
- If you are using two different components, you must use a fully scoped variable name (like `comp2.bcs3.A` for the area) when referencing the beam cross-section properties in the input fields of the beam interface.
- If a beam interface is added either after or at the same time as a Beam Cross Section interface, the only study type shown when adding a physics interface is **Stationary**. In this case, under **Custom Studies**, select **Preset Studies for Some Physics Interfaces** to find the other study types available for the beam analysis.

Values of Dependent Variables and Physics and Variables Selection in the *COMSOL Multiphysics Reference Manual*

For the Beam Cross Section interface in 2D, the cross sections are analyzed in the *xy*-plane. However, the beam interface uses a notation where the local *x*-axis is along the beam, and the cross section is described in a local *yz*-plane.

For the Beam Cross Section interface in 3D, the boundaries that represent cross sections can have arbitrary orientation. It is not related to the orientation of an actual beam.

In order to avoid confusion, the cross-section properties are described in the local x_1- and x_2-coordinates as shown in Figure 10-1 below.

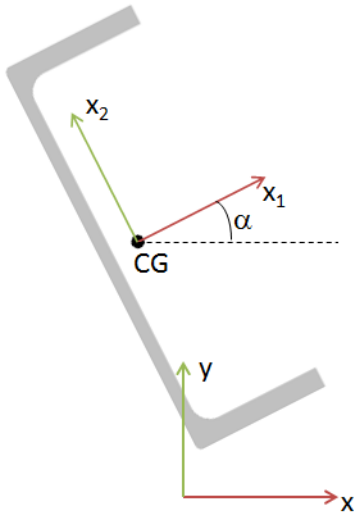


Figure 10-1: Local coordinate system and rotation angle.

When data is transferred to a beam interface, you must keep track of the coordinates that correspond to the local *y* and *z* directions.

	Channel Beam: Application Library path **Structural_Mechanics_Module/Verification_Examples/channel_beam**

The 3D version of the interface is based on the layered material technology. The reason is that the extra dimension inherent in layered materials is used to extrude the cross section when the interface is used to create a fully 3D representation of the stress state. Thus, there are significant conceptual differences between the 2D and 3D versions of this physics interface.

COMPUTING THE CROSS-SECTION DATA

You can compute the properties for several different cross sections in the same Beam Cross Section interface.

In a 2D component, the geometry of the cross sections is drawn in the *xy*-plane. In a 3D component, you can create one or more arbitrary work planes, in which the cross sections are drawn.

You need to attach material data to the cross section. The procedure is different in 2D and 3D.

The material properties can be assigned to the domain in a **Material** node, and then the option **From material** is used in the **Homogeneous Cross Section** node. Alternatively, you can also select **User defined**, and enter expressions for the material data manually.

Material data cannot be set locally in the **Homogeneous Cross Section** node. Usually, you would add a **Single Layer Material** node under **Materials** in the component to provide the material data. You can, however, also use a global **Layered Material**, which is then referenced from a **Layered Material Link** in the component. A boundary is not eligible for selection in the physics interface until a layered material has been assigned.

A layered material has, in addition the material data, three more inputs. The only one that may need to be changed is the **Thickness**, `lth`. It is not used as long as you are only computing cross-section properties, but if you are mapping results back from a beam interface, it should match the length of the beam.

If the section is not simply connected, add one Hole node for each internal hole. In that node, select all boundaries around the hole.

The default mesh density is tuned for thin-walled sections. For solid sections, an unnecessarily large model is obtained when using the default mesh.

The computed cross-section data is stored in the variables listed in Table 10-1:

TABLE 10-1: VARIABLES CONTAINING CROSS-SECTION DATA.

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
`bcs.A`	Area	Area (A)	Area

TABLE 10-1: VARIABLES CONTAINING CROSS-SECTION DATA.

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
`bcs.CGx`	Center of gravity, x-coordinate	Implicit, since it affects the positioning of the beam centerline.	Center of Gravity
`bcs.CGx`	Center of gravity, y-coordinate	As above	
`bcs.x1`	First coordinate in principal axes system	Implicit, since it can be used for determining stress evaluation locations	Local Coordinates
`bcs.x2`	Second coordinate in principal axes system	As above	
`bcs.I1`	Largest principal moment of inertia	Moment of inertia around local y/z-axis (Izz/Iyy)	Moments of Inertia
`bcs.I2`	Smallest principal moment of inertia	As above	
`bcs.Ixx`	Moment of inertia around x-axis	-	
`bcs.Iyy`	Moment of inertia around y-axis	-	
`bcs.Ixy`	Deviatoric moment of inertia in xy-system	-	
`bcs.rg`	Radius of gyration	-	
`bcs.alpha`	Angle from x-axis to first principal axis	Rotation of vector around beam axis (ϕ)	Directions of Principal Axes
`bcs.mu1`	Max shear stress factor, x1 direction	Max shear stress factor in local y/z direction ($\mu y/\mu z$)	Bending Shear Stresses
`bcs.mu2`	Max shear stress factor, x2 direction	As above	
`bcs.kappa1`	Shear correction factor, x1 direction	-	
`bcs.kappa2`	Shear correction factor, x2 direction	-	
`bcs.ei1`	Shear center location, first local coordinate	Distance to shear center in local y/z direction.	
`bcs.ei2`	Shear center location, second local coordinate	As above	

TABLE 10-1: VARIABLES CONTAINING CROSS-SECTION DATA.

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
bcs.J	Torsional constant	Torsional constant (J)	Torsion
bcs.Wt	Torsional section modulus	Torsional section modulus (Wt)	
bcs.Cw	Warping constant	-	Warping
bcs.Kw	Warping section modulus	-	
bcs.sw	Nonuniform torsion parameter	-	

COMPUTING DETAILED STRESSES

If you have a set of section forces (axial force, shear forces, bending moments, and twisting moments), it is possible to display the stresses it causes. To do this, enter the values in the **Section Forces** section. You can also add your own acceptance criteria by adding one or more Safety nodes.

The stresses are available in the variables listed in Table 10-2.

After changing data in this section, you do not need to compute the study for this physics interface again once is has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

TABLE 10-2: VARIABLES CONTAINING STRESS DISTRIBUTIONS.

VARIABLE	DESCRIPTION AND	LINK TO THEORY SECTION
bcs.sN	Stress from axial force	Axial Stress
bcs.sM1	Bending stress from moment around x1 axis	Bending Axial Stresses
bcs.sM2	Bending stress from moment around x2 axis	

TABLE 10-2: VARIABLES CONTAINING STRESS DISTRIBUTIONS.

VARIABLE	DESCRIPTION AND	LINK TO THEORY SECTION
`bcs.tT1x`	Shear stress from force in x1 direction, x-component	Bending Shear Stresses
`bcs.tT1y`	Shear stress from force in x1 direction, y-component	
`bcs.resT1`	Shear stress from force in x1 direction, resultant	
`bcs.tT2x`	Shear stress from force in x2 direction, x-component	
`bcs.tT2y`	Shear stress from force in x2 direction, y-component	
`bcs.resT2`	Shear stress from force in x2 direction, resultant	
`bcs.tMtx`	Shear stress from twisting moment, x-component	Torsional Shear Stresses
`bcs.tMty`	Shear stress from twisting moment, y-component	
`bcs.resMt`	Shear stress from twisting moment, resultant	
`bcs.mises`	von Mises equivalent stress	Equivalent Stress
`bcs.spN`	Principal stresses; N=1,2,3	
`bcs.INs`	N:th invariant of stress tensor; N=1,2,3	
`bcs.IINs`	N:th invariant of stress tensor deviator; N=2,3	
`bcs.thetaL`	Lode angle for stress tensor	

COMPUTING WARPING DISPLACEMENT

You can also compute the axial warping displacement by entering the axial twist of the beam in the **Twist** section. The theory is described under Warping. The axial displacement is stored in the variable `bcs.u`.

The Beam Cross Section Interface

The **Beam Cross Section (bcs)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for computing cross-section properties for beams. It can be also used for a detailed evaluation of stresses in a beam when the section forces to which the beam is subjected are known. It is mainly used in conjunction with The Beam Interface.

As input, you give a 2D sketch of the cross section. The computed results are: area, moments of inertia, center of gravity, principal axis directions, torsional rigidity, shear center location, shear correction factors, warping constant, and stress distributions for different load types.

- Using the Beam Cross Section Interface
- Theory for the Beam Cross Section Interface
- Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `bcs`.

DOMAIN SELECTION

For 2D components, select the domains that represent beam cross sections to be analyzed. The default setting is to include **All domains** in the model to define the cross section. To select specific domains, select **Manual** from the **Selection** list.

BOUNDARY SELECTION

For 3D components, select the boundaries that represent beam cross sections to be analyzed. The default setting is to include **All domains** in the model to define the cross section. To select specific domains, select **Manual** from the **Selection** list. Only boundaries that have an assigned a layered material can be selected.

BEAM PROPERTIES

For 3D components, this interface is based on the layered technology. The checkbox **Use all layers** is selected by default, and for all normal use, it does not have to be changed. The material selection is then implicit in the boundary selection.

DISCRETIZATION

Select the **Element order** — **Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**.

Homogeneous Cross Section

Add one **Homogeneous Cross Section** node for every cross section to be analyzed. One **Homogeneous Cross Section** node is automatically added when the physics interface is created.

MATERIAL PROPERTIES

Enter the material properties, **Young's modulus**, E and **Poisson's ratio**, ν. The default is to take the values **From material**. You can also select **User defined**, and enter expressions for the material data manually.

Currently only the value for ν is used, and it only influences the detailed distribution of shear stresses caused by a transversal load.

Even though it is possible to override material data from this section, there would seldom be a reason to do it. The material data is part of the definition of the layered material.

BEAM PROPERTIES

This section is only present in 3D, since in 3D the physics interface formally uses the layered technology. There is no need to make changes here.

SECTION FORCES

If you want to perform a detailed computation of the stress state in a beam cross section, enter the section forces.

Enter values or expressions for the following:

- **Axial force** N (SI unit: N).
- **Bending moment around 1-axis** M_1.
- **Shear force along 2-axis** T_2.
- **Bending moment around 2-axis** M_2.
- **Shear force along 1-axis** T_1.
- **Twisting moment** M_t.

- After changing data in this section, you do not need to compute the study for this physics interface again once is has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.
- If this interface is referenced in a Beam–Beam Cross Section Coupling, the inputs in this section are not accessible, since the cross section forces are transferred from a Beam interface or Pipe Mechanics interface.

TWIST

Enter a value or expression for the **Axial twist**, ϕ_{x1}. The twist is used in the computation of the axial warping displacement.

- After changing data in this section, you do not need to compute the study for this physics interface again once is has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.
- If this interface is referenced in a Beam–Beam Cross Section Coupling, the inputs in this section are not accessible, since the cross section forces are transferred from a Beam interface or Pipe Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Beam Cross Section > **Homogeneous Cross Section**

Ribbon

Physics tab with **Beam Cross Section** interface selected

Boundaries > **Homogeneous Cross Section (3D)**

Domains > **Homogeneous Cross Section (2D)**

Hole

Use the **Hole** subnode to define internal holes for a beam cross section. One **Hole** feature must be added for each internal hole in the parent Homogeneous Cross Section.

BEAM PROPERTIES

If you use the Beam Cross Section interface coupled to a Pipe Mechanics interface, the two interfaces have to agree on the type of cross section. Since circular pipes are very common in piping systems, they are to some extent treated as a special case in the Pipe Mechanics interface.

Select the **Pipe shape** — **Circular** or **Noncircular**. This setting is ignored for all cases, except in a coupled analysis with the Pipe Mechanics interface.

The settings for layers only present in 3D, since in 3D the physics interface formally uses the layered technology. There is no need to make changes here.

FLUID LOADS

The fluid loads are used for computing the stress from internal pressure. These inputs have a purpose analogous to the entries in the Section Forces and Twist sections in the **Homogeneous Cross Section** node.

Enter the **Internal pressure**, p_i, and the **External pressure**, p_o.

- After changing data in this section, you do not need to compute the study for this physics interface again once is has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.
- If this interface is referenced in a Beam–Beam Cross Section Coupling, the inputs in this section are not accessible, since the cross section forces are transferred from a Pipe Mechanics interface.

BOUNDARY SELECTION

You should select all boundaries around a certain hole from the **Selection** list (**Manual**). The **All boundaries** option is not relevant.

- Using the Beam Cross Section Interface
- Theory for the Beam Cross Section Interface

LOCATION IN USER INTERFACE

Context Menus

Beam Cross Section > **Homogeneous Cross Section** > **Hole**

Ribbon

Physics tab with **Homogeneous Cross Section** selected

Attributes > **Hole**

Safety

Use the **Safety** node to set up variables that can be used to check the risk of failure according to various criteria. Four different variables describing the failure risk will be defined as described in Table 10-3. You can add any number of **Safety** nodes.

TABLE 10-3: VARIABLES FOR SAFETY FACTOR EVALUATION.

VARIABLE	DESCRIPTION	CRITERION FULFILLED	CRITERION VIOLATED
Failure index, FI	For a linear criterion, this is the ratio between the computed value and the given limit.	FI<1	FI>1
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=1
Safety factor, SF	For a linear criterion, this is 1/FI.	SF>1	SF<1
Margin of safety, MoS	SF-1	MoS>0	MoS<0

BEAM PROPERTIES

This section is only present in 3D, since in 3D the physics interface formally uses the layered technology. There is no need to make changes here.

FAILURE MODEL

Select a **Failure Criterion** — **von Mises Isotropic**, **Tresca Isotropic**, **Rankine Isotropic**, **Bresler–Pister**, **Willam–Warnke**, or **Ottosen**.

- When **Failure Criterion** is **von Mises Isotropic**, enter **Tensile strength** σ_{ts}.
- When **Failure Criterion** is **Tresca Isotropic**, enter **Tensile strength** σ_{ts}.
- When **Failure Criterion** is **Rankine Isotropic**, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs}.
- When **Failure Criterion** is **Bresler–Pister Isotropic**, enter **Tensile strength** σ_{ts}, **Compressive strength** σ_{cs}, and **Biaxial compressive strength** σ_{bc}.
- When **Failure Criterion** is **Willam–Warnke Isotropic**, enter **Tensile strength** σ_{ts}, **Compressive strength** σ_{cs}, and **Biaxial compressive strength** σ_{bc}.

- When **Failure Criterion** is **Ottosen Isotropic**, enter the **Compressive strength** σ_{cs}, **Ottosen parameters** a and b, the **Size factor** k_1, and the **Shape factor** k_2.
- When **Failure Criterion** is **User defined**, you enter two expressions describing the **Failure criterion** $g(S)$, used in the failure index, and the **Safety factor** $s_f(S)$. As an example, if you would like to replicate the **von Mises Isotropic** criterion with tensile strength 350 MPa, you could enter $g(S)$ as `solid.mises/350[MPa]-1` and $s_f(S)$ as `350[MPa]/(solid.mises+eps)`.

For all input fields, the default is to take the value **From material**. Change to **User defined** to enter other values or expressions.

For a detailed description of the various criteria, see Safety Factor Evaluation in the *Structural Mechanics Theory* chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam Cross Section > **Homogeneous Cross Section** > **Safety**

Ribbon

Physics tab with Physics tab with **Homogeneous Cross Section** selected

Attributes > **Safety**

11

Truss

This chapter describes the Truss ( ) interface, which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Truss Interface

The **Truss (truss)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for modeling slender elements that can only sustain axial forces. It can be used for analyzing truss works where the edges are straight, or to model sagging cables like the deformation of a wire exposed to gravity. It is available in 3D and 2D. Geometric nonlinearity can be taken into account.

The Truss interface can also be used for modeling springs and dashpots.

The default material model is Linear Elastic Material. With the Nonlinear Structural Materials Module, you can also model Plasticity and Shape Memory Alloys.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, these default nodes are also added to the Model Builder: **Linear Elastic Material**, **Cross-Section Data**, **Free** (a condition where points are free, with no loads or constraints), **Straight Edge Constraint** (to ensure that the points lie on a straight line between the endpoints of the edge or boundary), and **Initial Values**. Then, from the **Physics** toolbar, you can add other nodes that implement, for example, loads and constraints. You can also right-click **Truss** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `truss`.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more

efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

ENERGY DISSIPATION

From the **Store dissipation** list, select **Individual contributions**, **Total**, or **Off**.

Select **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Select **Total** to accumulate all the dissipative processes into one variable.

DISCRETIZATION

The default is to use linear shape functions. If the truss elements share an edge with another structural mechanics interface, you may want to use quadratic shape functions instead, in order to get displacement compatibility.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field u** which has two components (`u`, `v`) in 2D and three components (`u`, `v`, and `w`) in 3D. The name can be changed but the names of fields and dependent variables must be unique within a model.

- Boundary, Edge, Point, and Pair Nodes for the Truss Interface, Explicit Dynamics
- Theory for the Truss Interface

Boundary, Edge, Point, and Pair Nodes for the Truss Interface, Explicit Dynamics

The Truss Interface has these boundary, edge, point, and pair nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Truss interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases. The submenus at the **Edge** level (3D) or **Boundary** level (2D) are

- **Material Models**
- **Line and Volume Loads**
- **Mass, Spring, and Damper**
- **Line Constraints**

The submenus at the **Point** level are

- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Antisymmetry
- Cross-Section Data
- Edge Load
- External Stress
- Hygroscopic Swelling
- Initial Stress and Strain
- Linear Elastic Material
- Pinned
- Point Mass
- Set Variables
- Spring-Damper Material
- Straight Edge Constraint
- Symmetry
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Linear Elastic Material)

These nodes are described for the Solid Mechanics interface

- Activation
- Added Mass
- Adiabatic Heating
- Average Rotation
- Base Excitation
- Damping
- Free
- Gravity
- Harmonic Perturbation
- Initial Values
- Linearly Accelerated Frame[1]
- Phase
- Plasticity
- Point Load
- Point Load, Free
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rotating Frame[1]
- Shape Memory Alloy
- Spring Foundation
- Spring–Damper

[1] This is selected from the **Line and Volume Loads** submenu for this interface.

If there are subsequent constraints on the same geometrical entity, the last one takes precedence.

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Cross-Section Data

Use the **Cross-Section Data** node to describe the section for the truss elements on the selected edges.

CROSS-SECTION DEFINITION

Select a **Section type** — **Rectangle**, **Box**, **Circular**, **Pipe**, **H-profile**, **U-profile**, **T-profile**, **C-profile**, **Hat**, or **User defined**. Then go to the relevant section below to continue

defining the section. Each **Section type** in the table below also has a figure showing the section and its defining dimensions.

	For equations and a figure see: • Rectangular Section • Box Section • Circular Section • Pipe Section • H-Profile Section • U-Profile Section • T-Profile Section • C-Profile Section • Hat Section

Rectangle

Enter values or expressions for the following.

- **Width in local y direction** h_y
- **Width in local z direction** h_z

Box

Enter values or expressions for the following.

- **Width in local y direction** h_y
- **Width in local z direction** h_z
- **Wall thickness in local y direction** t_y
- **Wall thickness in local z direction** t_z

Circular

Enter a value or expression for the **Diameter** d_o.

Pipe

Enter values or expressions for the following.

- **Outer diameter** d_o
- **Inner diameter** d_i

H-profile, U-profile, or T-profile

Enter values or expressions for the following.

- **Section height** h_y
- **Flange width** h_z
- **Flange thickness** t_y
- **Web thickness** t_z

C-profile

Enter values or expressions for the following.

- **Flange width** h_y
- **Flange thickness** t_z
- **Web height** h_z
- **Web thickness** t_y
- **Lip width** w_l
- **Lip thickness** t_l

Hat

Enter values or expressions for the following.

- **Crown width** h_v
- **Crown thickness** t_c
- **Web height** h_w
- **Web thickness** t_w
- **Flange width** w_f
- **Flange thickness** t_f

User Defined

Enter the cross-section **Area** A.

LOCAL BUCKLING

Optionally, you can also provide data that makes it possible to assess the risk of local buckling of the individual bars in the structure.

Enter the **Smallest principal moment of inertia**, $I_{\min}$. This is the area moment of inertia in the weakest direction of the cross section. This input is only required for a user-defined cross section. For one of the predefined cross sections, the moment of inertia is computed from the geometrical properties.

Enter the **Effective length factor**, K. This is a multiplier to the physical length of the bars, intended for compensating for end conditions. The default value, 1, corresponds to a pure Euler 2 buckling case.

See also Local Buckling.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Cross-Section Data**
Truss, Explicit Dynamics > **Cross-Section Data**

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Truss** > **Cross-Section Data**

Straight Edge Constraint

The **Straight Edge Constraint** controls the addition of an additional constraint, forcing the edge to be straight. The default is to add this constraint to all edges unless linear shape functions have been chosen in the Discretization section of the interface. Using this additional constraint removes the need to use a mesh with only one element per edge. Internal nodes make the model singular because the truss element only has stiffness in the axial direction in a geometrically linear analysis. The same problem appears when using higher-order shape functions, even if there is only one element along an edge since there are internal nodes.

In the case of geometric nonlinearity, there is a stiffness in the transverse directions as long as the axial force is tensile.

The additional constraints increase the solution time, especially for large 3D and transient problems. The default mesh, when using **Physics-controlled mesh** in the **Mesh** node, is to use one element per edge only, so that the extra constraints are not needed unless the shape functions are of higher order.

STRAIGHT EDGE CONSTRAINT

Select a **Automatic**, **On**, or **Off**. The **Automatic** option adds the straight edge constraints on the selected edges except when linear shape functions are applied to a single mesh element per edge. This is the default setting.

Select **Off** to always disable the straight edge constraints.

The **Straight Edge Constraint** is not applicable to edges that are embedded in a solid domain through the **Embedded Reinforcement** multiphysics coupling.

See also Theory for Straight Edge Constraint.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Line Constraints** > **Straight Edge Constraint**

Truss, Explicit Dynamics > **Line Constraints** > **Straight Edge Constraint**

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Line Constraints** > **Straight Edge Constraint**

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic truss element, and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- Thermal Expansion (for Linear Elastic Material)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- Damping
- Plasticity
- Activation
- Safety

LINEAR ELASTIC MATERIAL

Define the linear elastic material properties. These settings are the same as described under Linear Elastic Material for the Beam interface.

GEOMETRIC NONLINEARITY

The settings in this section control the overall kinematics, the definition of the strain decomposition, and the behavior of inelastic contributions, for the material.

Select a **Formulation** — **From study step**, **Total Lagrangian**, or **Geometrically linear** to set the kinematics of the deformation and the definition of strain. When **From study step** is selected, the study step controls the kinematics and the strain definition.

When **From study step** is selected, a total Lagrangian formulation for large strains is used when the **Include geometric nonlinearity** checkbox is selected in the study step. If the checkbox is not selected, the formulation is geometrically linear, with a small strain formulation.

To have full control of the formulation, select either **Total Lagrangian**, or **Geometrically linear**. When **Total Lagrangian** is selected, the physics will force the **Include geometric nonlinearity** checkbox in all study steps.

When inelastic deformations are present, such as for plasticity, the elastic strain can be obtained in different ways: using additive decomposition of strains or logarithmic stretches, or using multiplicative decomposition of deformation gradients.

Select a **Strain decomposition** — **Automatic**, **Additive**, **Logarithmic**, or **Multiplicative** to decide how the inelastic deformations are treated. This option is not available when the formulation is set to **Geometrically linear**.

- When **Automatic** is selected, a multiplicative or additive decomposition is used with a total Lagrangian formulation, depending on the **Include geometric nonlinearity** checkbox status in the study step.
- Select **Additive** to force an additive decomposition of strains.

- Select **Logarithmic** to force an additive decomposition of logarithmic stretches. This option is only visible if **Formulation** is set to **Total Lagrangian**.
  - Select a **Method** — **Analytic** or **Padé** to decide how the logarithm of the right stretch tensor is computed.
- Select **Multiplicative** to force a multiplicative decomposition of deformation gradients. This option is only visible if **Formulation** is set to **Total Lagrangian**.

The **Logarithmic** strain decomposition is available for **Linear Elastic** materials in the Solid Mechanics and Solid Mechanics, Explicit Dynamics interfaces.

The **Strain decomposition** input is only visible for material models that support both additive and multiplicative decomposition of the deformation gradient.

- There are some cases when a small strain formulation could be useful for a certain domain, even though the study step is geometrically nonlinear. One such case is contact analysis, where the study step is always geometrically nonlinear, but where a geometrically linear formulation is sufficient for the material.
- When a multiplicative decomposition is used, the order of the subnodes matters. The inelastic deformations are assumed to occur in the same order as the subnodes appear in the model tree.

For more information, see

- Lagrangian Formulation, Deformation Measures, and Inelastic Strain Contributions in the *Structural Mechanics Theory* chapter.
- Modeling Geometric Nonlinearity in the *Structural Mechanics Modeling* chapter.
- Study Settings in the *COMSOL Multiphysics Reference Manual*.

ENERGY DISSIPATION

Select how to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or other dissipative processes.

Select how to **Store dissipation** — **From physics interface**, **Individual contributions**, **Total**, **Domain ODEs (legacy)**, or **Off**.

The option **Domain ODEs (legacy)** is not available in the interfaces intended for time-explicit dynamic analysis.

Use **From physics interface** to treat the dissipative processes as specified in the settings of the physics interface, see for instance Energy Dissipation in the Solid Mechanics interface.

Use **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Use **Total** to accumulate all the dissipative processes into one common variable.

Use **Domain ODEs** to accumulate the dissipative processes into ODE variables instead of internal state variables.

- Dissipated Energy
- Energy Variables

QUADRATURE SETTINGS

Select the **Reduced integration** checkbox to reduce the number of integration points for the weak contribution from the material.

The default is that the **Reduced integration** checkbox is cleared in the Truss interface and selected in the Truss, Explicit Dynamics interface.

See also Reduced Integration and Hourglass Stabilization in the *Structural Mechanics Theory* chapter and Using Reduced Integration in the *Structural Mechanics Modeling* chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Material Models** > **Linear Elastic Material**

Truss, Explicit Dynamics > **Material Models** > **Linear Elastic Material**

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Material Models** > **Linear Elastic Material**

Set Variables

Use the **Set Variables** subnode to Plasticity of a Linear Elastic Material to reset plasticity variables according to a **Setting condition** that you define. When the **Setting condition** is satisfied, the plasticity variables are reset to the specified values.

SET VARIABLES

Enter the **Setting condition**. This is a Boolean expression that will determine when the plastic variables are reset.

From the **Equivalent plastic strain** list, select **Do not set** or **User defined**. The default **User defined** value is zero. Depending on the type of plasticity model, set additional plasticity variables.

From the **Plastic axial strain** list, select **Do not set** or **User defined**. The default **User defined** values is zero.

If **Armstrong–Frederick** or **Chaboche** is selected from the Kinematic Hardening list, specify the values for the **Back strain**. From the **Back strain** list, select **Do not set** or **User defined**. The default **User defined** values is zero.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Material Models** > **Plasticity** > **Set Variables**
Truss, Explicit Dynamics > **Material Models** > **Plasticity** > **Set Variables**

Ribbon

Physics tab with **Plasticity** node selected in the model tree:

Attributes > **Set Variables**

Thermal Expansion (for Linear Elastic Material)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

From the **Temperature** T list, select an existing temperature variable from another physics interface, if any such temperature variables exist, or select **User defined** to enter a value or expression for the temperature.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*
- Temperature-Dependent Material Data

THERMAL EXPANSION PROPERTIES

Select an **Input type** to specify how the thermal strain is defined. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

where α is the secant coefficient of thermal expansion, which can be temperature dependent. As a default, the **Common model input**. You can also select **User defined** to enter a value or expression.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. Select **User defined** to enter an expression for the coefficient of thermal expansion or thermal strain.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Linear Elastic Material** > **Thermal Expansion**

Truss, Explicit Dynamics > **Linear Elastic Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Thermal Expansion (for Shape Memory Alloy)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

The current **Temperature** T of the alloy is defined in the parent **Shape Memory Alloy** node.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

THERMAL EXPANSION PROPERTIES

Select an **Input type** to specify how the thermal strain is defined. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

here, $\alpha = \xi_{\text{A}}\alpha_{\text{A}} + \xi_{\text{M}}\alpha_{\text{M}}$ is the volume average secant coefficient of thermal expansion, and ξ_{A} and ξ_{M} are the austenite and martensite volume fractions. The default **Secant coefficient of thermal expansion for Austenite** α_{A} and **Secant coefficient of thermal expansion for Martensite** α_{M} use values **From material**.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where $\alpha_t = \xi_A\alpha_{tA} + \xi_M\alpha_{tM}$ is the volume average tangential coefficient of thermal expansion. The default **Tangent coefficient of thermal expansion for Austenite** α_{tA} and **Tangent coefficient of thermal expansion for Martensite** α_{tM} use values **From material**.

When **Input type** is **Thermal strain**, enter the **Thermal strain for Austenite** dL_A and the **Thermal strain for Martensite** dL_M as function of temperature explicitly. Then the thermal strain of the alloy is computed from $dL = \xi_A dL_A + \xi_M dL_M$.

In all three cases, the default is to take values **From material**. Select **User defined** to enter a value or expression for the coefficient of thermal expansion or thermal strain.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Shape Memory Alloy** > **Thermal Expansion**

Ribbon

Physics tab with **Shape Memory Alloy** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. The temperature is assumed to be constant over the cross section of the truss element.

The settings for the Truss interface are the same as described for the Beam interface (excluding the hygroscopic bending options). See Hygroscopic Swelling in the documentation for the Beam interface.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Linear Elastic Material** > **Hygroscopic Swelling**

Truss, Explicit Dynamics > **Linear Elastic Material** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to the Linear Elastic Material, in order to specify the stress or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense, but rather a contribution to the constitutive relation

In many cases **Initial Stress and Strain** and External Stress are interchangeable when prescribing stresses, but you can find some more options in the latter.

INITIAL STRESS AND STRAIN

Enter an **Initial axial strain** e_{ni} and **Initial axial stress** σ_{ni}.

For details about initial stresses and strains, see Inelastic Strain Contributions and Initial Stress and Strain.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Linear Elastic Material** > **Initial Stress and Strain**

Truss, Explicit Dynamics > **Linear Elastic Material** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Initial Stress and Strain**

External Stress

You can add the **External Stress** subnode to the Linear Elastic Material, in order to specify an additional stress contribution which is not part of the constitutive relation. The external stress can be added to the total stress tensor, or act only as an extra load contribution.

In many cases **External Stress** and Initial Stress and Strain are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor** or **Axial stress**.

- When **Stress tensor** is selected, you enter the external stress in the form of Second Piola–Kirchhoff stress tensors. In the **External stress tensor** list, stress tensors announced by any physics interface will be shown, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** S_{ext} as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the global coordinate system, and are projected onto the tangential direction of the truss element. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** to determine the effect of the contribution.
- When **Axial stress** is selected, you enter a value or an expression for the **Axial stress** $S_{\text{n,ext}}$. Choose a **Contribution type** — **Add to stress tensor**, **Load contribution only**, or **Residual stress** or to determine the effect of the contribution.

Selecting a stress tensor announced by the same physics interface as where the **External Stress** node is added, will result in an error (‘`Circular variable dependency detected`’). This operation would imply that the computed stress depends on itself.

For theory, see External Stress.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Linear Elastic Material** > **External Stress**

Truss, Explicit Dynamics > **Linear Elastic Material** > **External Stress**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **External Stress**

Spring-Damper Material

Use a **Spring-Damper Material** to model — between two points — an elastic spring, a viscous damper, or both. By adding a Truss interface with this material model, you can add springs and dashpots to any other structural mechanics physics interface.

SPRING-DAMPER

Select a **Spring type** — **Spring constant** or **Force as function of extension**.

- For **Spring constant** enter a value for the spring constant k.
- For **Force as function of extension** enter an expression for the spring force F_s. The expression must be a function of the extension of the spring. The built-in variable for the spring extension has the form `<physicsName>.<SpringNodeTag>.dl`, for example `truss.spd1.dl`. The default expression is `(1[N/m])*truss.spd1.dl`, which corresponds to a linear spring with the stiffness 1 N/m.

To add viscous damping in a dynamic analysis, enter a value or expression for the **Damping coefficient** c.

To add loss factor damping, enter a value for the **Loss factor damping** η.

FREE LENGTH

The free length is the distance between the connection points when there is no force in the spring. Select an option from the list — **Specify initial extension** or **Specify free length**.

- For **Specify initial extension** enter a value for Δl_0. The free length is computed as $l_f = l_0 - \Delta l_0$, where l_0 is the initial distance between the connection points.
- For **Specify free length** enter a value for l_f.

ACTIVATION CONDITIONS

Select a **Spring action** — **Bidirectional**, **Tension only**, or **Compression only**.

Select the **Deactivation** checkbox to enter a **Deactivation expression** i_{dac}. The expression is treated as a Boolean expression, so that when it is evaluated to a nonzero value, the spring or damper is deactivated.

Select the **Permanently deactivate** checkbox if the spring is supposed to be removed permanently from the simulation once the deactivation condition is fulfilled for the first time.

As an example, if the spring should break at a certain extension, you can write an expression like `truss.spd1.dl>0.12[m]`, and select the **Permanently deactivate** checkbox. If the checkbox is not selected, the spring would become active again when its connection points come close enough to each other.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Material Models** > **Spring-Damper Material**

Ribbon

Physics tab with **Truss** selected:

Edges > **Material Models** > **Spring-Damper Material**

Pinned

The **Pinned** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that makes the edge, boundary, or point fixed; that is, the displacements are zero in all directions.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Line Constraints** > **Pinned** (Line)
Truss > **Pinned** (Point)
Truss, Explicit Dynamics > **Line Constraints** > **Pinned** (Line)
Truss, Explicit Dynamics > **Pinned** (Point)

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Line Constraints** > **Pinned**
Points > **Truss** > **Pinned**

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Pinned** or **Prescribed Displacement**) to prescribe a deformation of the constraint caused by changes in temperature of the surroundings. This makes it possible to reduce stresses caused by the boundary conditions.

The thermal strain depends on the coefficient of thermal expansion α, the temperature T, and the strain-free reference temperature T_{ref} as

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from edge** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the edge itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** α. As a default, values **From material** are used. This requires that a material has been assigned to the points where the constraint is active.
  For **User defined** enter the coefficient of thermal expansion α.
- A value or expression for the **Volume reference temperature** T_{ref} which is the temperature at which there are no thermal displacements at the constraints.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Pinned** > **Thermal Expansion**
Truss > **Prescribed Displacement** > **Thermal Expansion**
Truss, Explicit Dynamics > **Pinned** > **Thermal Expansion**
Truss, Explicit Dynamics > **Prescribed Displacement** > **Thermal Expansion**

Ribbon

Physics tab with **Pinned** or **Prescribed Displacement** node selected in the model tree:

Attributes > **Thermal Expansion**

Symmetry

The **Symmetry** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that defines a symmetry edge, boundary, or point.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. The coordinate system is used in conjunction with the **Axis to use as normal direction** setting.

SYMMETRY

Select an **Axis to use as normal direction**. This specifies the direction of the normal to the symmetry plane. Select **1**, **2**, or **3** for the first, second, or third axis in the selected coordinate system.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_{n}. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.truss.sym1.un`.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Line Constraints** > **Symmetry** (Line)
Truss > **More Constraints** > **Symmetry** (Point)
Truss, Explicit Dynamics > **Line Constraints** > **Symmetry** (Line)

Truss, Explicit Dynamics > **More Constraints** > **Symmetry** (Point)

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Line Constraints** > **Symmetry**
Points > **More Constraints** > **Symmetry**

Antisymmetry

The **Antisymmetry** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that defines an antisymmetry edge, boundary, or point.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. The coordinate system is used in conjunction with the **Axis to use as normal direction** setting.

ANTISYMMETRY

Select an **Axis to use as normal direction**. This specifies the direction of the normal to the antisymmetry plane.

- For 2D models, select **1** or **2** for the first or second axis, respectively.
- For 3D models, select **1**, **2**, or **3** for the first, second, or third axis, respectively.

When

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

	Symmetry Constraints

	You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Line Constraints** > **Antisymmetry** (Line)
Truss > **More Constraints** > **Antisymmetry** (Point)
Truss, Explicit Dynamics > **Line Constraints** > **Antisymmetry** (Line)
Truss, Explicit Dynamics > **More Constraints** > **Antisymmetry** (Point)

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Line Constraints** > **Antisymmetry**
Points > **More Constraints** > **Antisymmetry**

Edge Load

Add an **Edge Load** as a force distributed along an edge (3D models) or boundary (2D models).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the model. Prescribed loads are specified along the axes of this coordinate system. See Coordinate Systems and in the *COMSOL Multiphysics Reference Manual*.

FORCE

Select a **Load type** — **Force per reference volume**, **Force per deformed volume**, **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for the components.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference volume	$\mathbf{f}_V$	N/m^3	edges boundaries	3D (x, y, z) 2D (x, y)
Force per deformed volume	$\mathbf{f}_v$	N/m^3	edges boundaries	3D (x, y, z) 2D (x, y)
Force per reference length	$\mathbf{f}_L$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	edges boundaries	3D (x, y, z) 2D (x, y)

- When **Force per reference volume** is selected, the given load is multiplied by the cross-section area. This option is useful for modeling body loads like gravity or centrifugal loads.
- After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the Truss interface with another physics interface, it is also possible to choose a predefined load from this list.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Line and Volume Loads** > **Edge Load**

Truss, Explicit Dynamics > **Line and Volume Loads** > **Edge Load**

Ribbon

Physics tab with **Truss** or **Truss, Explicit Dynamics** selected:

Edges > **Line and Volume Loads** > **Edge Load**

Point Mass

Use the **Point Mass** node to add a discrete mass that is concentrated at a point.

POINT MASS

Enter a **Point mass** m.

DAMPING

Enter a **Mass damping parameter** α_{dM}. This is the mass proportional term of a Rayleigh damping.

Select the **Exclude contribution** checkbox to switch off the loads generated by this node when the frame is accelerated using a Gravity, Rotating Frame, Linearly Accelerated Frame, or Base Excitation feature. The setting will also determine whether the node will contribute when computing mass properties and participation factors.

For more information, see Computing Mass Properties and Participation Factors.

LOCATION IN USER INTERFACE

Context Menus

Truss > **Mass, Spring, and Damper** > **Point Mass**
Truss, Explicit Dynamics > **Mass, Spring, and Damper** > **Point Mass**
Wire > **Mass, Spring, and Damper** > **Point Mass**

Ribbon

Physics tab with **Truss**; **Truss, Explicit Dynamics**; or **Wire** selected:

Points > **Mass, Spring, and Damper** > **Point Mass**

12

Truss, Explicit Dynamics

This chapter describes the Truss, Explicit Dynamics ( ) interface, which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Truss, Explicit Dynamics Interface

The **Truss, Explicit Dynamics (truss)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for general time explicit analysis of slender elements that can only sustain axial forces. It can be used for analyzing truss works where the edges are straight, or to model sagging cables like the deformation of a wire exposed to gravity. It is available in 3D and 2D.

Explicit Dynamics in the COMSOL Multiphysics Reference Manual.

This interface is similar to the The Truss Interface, upon which it is based. There are a number of specializations for explicit dynamics, but the interface can also be used with most other study types. The most important use case for that is to provide initial conditions for a time-explicit analysis.

Some important differences, when compared to the Truss interface are:

- The only top level material models available is **Linear Elastic Material**. Most its subnodes, such as **Plasticity**, **Damping**, or **Thermal Expansion** are available.
- Some other features are not available., most noticeably **Spring-Damper**. A detailed list of the available features can be found in the section Boundary, Edge, Point, Global and Pair Nodes for the Truss, Explicit Dynamics Interface below.
- The **Straight Edge Constraint** node is not added by default.
- Only linear shape functions are supported. Reduced integration is used as a default
- Weak constraints are not available.
- In cases where there are smaller differences between the settings for a certain feature, for example different options is a list, it is noted in the documentation of that feature.

Note that you can move nodes between the Truss and the Truss, Explicit Dynamics interfaces simply by drag-and-drop.

The default material model is Linear Elastic Material. With the Nonlinear Structural Materials Module, you can also model Plasticity.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, these default nodes are also added to the Model Builder: **Linear Elastic Material**, **Cross-Section Data**, **Free** (a condition where points are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, you can add other nodes that implement, for example, loads and constraints. You can also right-click **Truss, Explicit Dynamics** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `truss`.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

ENERGY DISSIPATION

From the **Store dissipation** list, select **Individual contributions**, **Total**, or **Off**.

Select **Individual contributions** to treat each dissipative process independently. Selecting this option gives a more flexible implementation for problems where dissipation occurs at different time scales, and you want to distinguish each phenomenon separately.

Select **Total** to accumulate all the dissipative processes into one variable.

DISCRETIZATION

In the Truss, Explicit Dynamics interface, the only available discretization is linear shape functions.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field u** which has two components (`u`, `v`) in 2D and three components (`u`, `v`, and `w`) in 3D. The name can be changed but the names of fields and dependent variables must be unique within a model.

- Boundary, Edge, Point, Global and Pair Nodes for the Truss, Explicit Dynamics Interface
- Theory for the Truss Interface

Boundary, Edge, Point, Global and Pair Nodes for the Truss, Explicit Dynamics Interface

Most of the features in the Truss Explicit Dynamics interface are identical to the same features in the Solid Mechanics interface, and are thus documented there.

The Truss, Explicit Dynamics interface has these boundary, edge, point, and pair nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Truss interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases. The submenus at the **Edge** level (3D) or **Boundary** level (2D) are

- **Material Models**
- **Line and Volume Loads**
- **Mass, Spring, and Damper**

- **Line Constraints**
- **More**

The submenus at the **Point** level are

- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes are described for the Truss interface:

- Antisymmetry
- Cross-Section Data
- Edge Load
- External Stress
- Hygroscopic Swelling
- Initial Stress and Strain
- Linear Elastic Material
- Pinned
- Point Mass
- Set Variables
- Straight Edge Constraint
- Symmetry
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Linear Elastic Material)

These nodes are described for the Solid Mechanics, Explicit Dynamics interface::

- Mass Scaling
- Prescribed Displacement (on Domains)

These nodes are described for the Solid Mechanics interface

- Activation
- Added Mass
- Adiabatic Heating
- Damping
- Free
- Gravity
- Harmonic Perturbation
- Initial Values
- Linearly Accelerated Frame[1]
- Phase
- Plasticity
- Point Load
- Predeformation
- Prescribed Displacement (on points, edges, and boundaries)
- Rotating Frame[1]
- Safety
- Spring Foundation

[1] This is selected from the **Line and Volume Loads** submenu for this interface.

If there are subsequent constraints on the same geometrical entity, the last one takes precedence.

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

13

Wire

This chapter describes the Wire interface ( ), which is found under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Wire Interface

The **Wire (wire)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for modeling cables or wires that can only sustain tensile axial forces. Geometric nonlinearity is always active for this interface, since the transverse stiffness of a wire is caused by its prestress.

The default material model is Elastic Wire.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, these default nodes are also added to the Model Builder: **Elastic Wire**, **Free** (a condition where points are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, you can add other nodes that implement, for example, loads and constraints. You can also right-click **Wire** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `wire`.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

DISCRETIZATION

The only available discretization is linear shape functions.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field u** which has two components (u, v) in 2D and three components (u, v, and w) in 3D. The name can be changed but the names of fields and dependent variables must be unique within a model.

- Boundary, Edge, Point, and Pair Nodes for the Wire Interface
- Edge Load
- Theory for the Wire Interface

Vibrating String: Application Library path **Structural_Mechanics_Module/Verification_Examples/vibrating_string**

Boundary, Edge, Point, and Pair Nodes for the Wire Interface

The Wire Interface has these boundary, edge, point, and pair nodes which are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Wire interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases. The submenus at the **Edge** level (3D) or **Boundary** level (2D) are:

- **Line and Volume Loads**
- **Mass, Spring, and Damper**
- **Line Constraints**

The submenus at the **Point** level are

- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Antisymmetry
- Edge Load
- Elastic Wire
- Hygroscopic Swelling
- Initial Stress and Strain
- Pinned
- Symmetry
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Elastic Wire)

These nodes are described for the Solid Mechanics, Truss, or Beam interface:

- Activation
- Added Mass
- Average Rotation
- Base Excitation
- Damping
- Free
- Gravity
- Harmonic Perturbation
- Inertia Relief
- Initial Values
- Linearly Accelerated Frame[1]
- Phase
- Point Load
- Point Load, Free
- Point Mass
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rotating Frame[1]
- Spring Foundation
- Spring–Damper

[1] This is selected from the **Line and Volume Loads** submenu for this interface.

If there are subsequent constraints on the same geometrical entity, the last one takes precedence.

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Elastic Wire

The **Elastic Wire** node adds the equations for a linear elastic wire element, and an interface for defining the elastic material properties.

By adding the following subnodes to the **Elastic Wire** node you can incorporate many other effects:

- Thermal Expansion (for Elastic Wire)
- Hygroscopic Swelling
- Initial Stress and Strain
- Damping
- Activation

WIRE PROPERTIES

Enter the **Axial Stiffness**, k_A, and **Mass per reference length**, ρ_L.

You can use parameters in **Compressive stiffness** to tune how the stiffness is reduced under compression.

Enter the **Compressive stiffness reduction factor**, β. The asymptotic stiffness under significant compressive strains is βk_A.

Enter the **Transition region strain limit**, ε_c. The stiffness is reduced from the full stiffness at zero strain, to the value βk_A at the negative strain ε_c. An exponential decay function is used.

For details, see also Stress–Strain Relation

CROSS-SECTION DATA

Enter the **Area**, A. The cross-section area as such is not essential to the analysis. It is only used during result presentation. Note that wires sometimes have

nonhomogeneous cross sections, so the definition of a cross-section area may not be obvious.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Elastic Wire**

Ribbon

Physics tab with **Wire** selected:

Edges > **Elastic Wire**

Thermal Expansion (for Elastic Wire)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

From the **Temperature** T list, select an existing temperature variable from another physics interface, if any such temperature variables exist, or select **User defined** to enter a value or expression for the temperature.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*
- Temperature-Dependent Material Data

THERMAL EXPANSION PROPERTIES

Select an **Input type** to specify how the thermal strain is defined. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

where α is the secant coefficient of thermal expansion, which can be temperature dependent. As a default, the **Common model input**. You can also select **User defined** to enter a value or expression.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. Select **User defined** to enter an expression for the coefficient of thermal expansion or thermal strain.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Linear Elastic Material** > **Thermal Expansion**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. The temperature is assumed to be constant over the cross section of the wire element.

The settings for the Wire interface are the same as described for the Beam interface (excluding the hygroscopic bending options). See Hygroscopic Swelling in the documentation for the Beam interface.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Elastic Wire** > **Hygroscopic Swelling**

Ribbon

Physics tab with **Elastic Wire** node selected in the model tree:

Attributes > **Hygroscopic Swelling**

Initial Stress and Strain

You can add the **Initial Stress and Strain** subnode to the Elastic Wire, in order to specify the force or strain state in the structure before applying any constraint or load. The values given are not initial values in the mathematical sense, but rather a contribution to the constitutive relation

INITIAL STRESS AND STRAIN

Enter an **Initial axial strain** e_{n0} and **Initial axial force** N_i.

For details about initial stresses and strains, see Stress–Strain Relation.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Elastic Wire** > **Initial Stress and Strain**

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes > **Initial Stress and Strain**

Pinned

The **Pinned** node adds an edge (3D), boundary (2D), or point (2D and 3D) condition that makes the edge, boundary, or point fixed; that is, the displacements are zero in all directions.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Line Constraints** > **Pinned** (Line)
Wire > **Pinned** (Point)

Ribbon

Physics tab with **Wire** selected:

Edges > **Line Constraints** > **Pinned**
Points > **Wire** > **Pinned**

Thermal Expansion (for Constraints)

Add the **Thermal Expansion** subnode to a constraint (**Pinned** or **Prescribed Displacement**) to prescribe a deformation of the constraint caused by changes in temperature of the surroundings. This makes it possible to reduce stresses caused by the boundary conditions.

The thermal strain depends on the coefficient of thermal expansion α, the temperature T, and the strain-free reference temperature T_{ref} as

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from edge** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the edge itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for **Temperature** T, specifying the temperature distribution of the surrounding material. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** α. As a default, values **From material** are used. This requires that a material has been assigned to the points where the constraint is active.
  For **User defined** enter the coefficient of thermal expansion α.
- A value or expression for the **Volume reference temperature** T_{ref} which is the temperature at which there are no thermal displacements at the constraints.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.

- Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.
- Thermal Expansion of Constraints in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Pinned** > **Thermal Expansion**

Wire > **Prescribed Displacement** > **Thermal Expansion**

Ribbon

Physics tab with **Pinned** or **Prescribed Displacement** node selected in the model tree:

Attributes > **Thermal Expansion**

Symmetry

The **Symmetry** node adds a point condition that defines a symmetry point.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. The coordinate system is used in conjunction with the **Axis to use as normal direction** setting.

SYMMETRY

Select an **Axis to use as normal direction**. This specifies the direction of the normal to the symmetry plane. Select **1**, **2**, or **3** for the first, second, or third axis in the selected coordinate system.

NORMAL DIRECTION CONDITION

You can allow a symmetry plane to move along its normal direction. This can be used to model some situations where you want a plane to remain strictly planar but still relax the property of it being fixed.

From the list, select **No displacement**, **Free Displacement**, **Prescribed force**, or **Prescribed displacement**.

The value **No displacement** gives a standard symmetry condition.

Select **Free Displacement** to allow the symmetry plane to translate in the normal direction. The displacement is determined by the criterion that there is no resulting reaction force in the normal direction.

Select **Prescribed force** to prescribe the total reaction force acting on the direction normal to the symmetry plane. Enter the **Normal force** F_{n}. The force is defined as positive when acting along the outward normal of the symmetry plane. Setting the prescribed force to zero gives the same effect as using **Free Displacement**.

Select **Prescribed displacement** to prescribe the displacement in the direction normal to the symmetry plane. Enter the **Normal displacement** u_{n0}. Setting the prescribed displacement to zero gives the same effect as using **No displacement**.

- Using **Free Displacement**, **Prescribed force**, or **Prescribed displacement** is only meaningful if the geometry selection corresponds to a single symmetry plane.
- When using **Free Displacement** or **Prescribed force**, an extra global degree of freedom is added for determining the displacement in the normal direction. This degree of freedom will have a name of the type `<component>.<interface>.<symmetry_tag>.un`, for example `comp1.wire.sym1.un`.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Wire > **More Constraints** > **Symmetry**

Ribbon

Physics tab with **Wire** selected:

Points > **More Constraints** > **Symmetry**

Antisymmetry

The **Antisymmetry** node adds a point condition that defines an antisymmetry point.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. The coordinate system is used in conjunction with the **Axis to use as normal direction** setting.

ANTISYMMETRY

Select an **Axis to use as normal direction**. This specifies the direction of the normal to the antisymmetry plane.

- For 2D models, select **1** or **2** for the first or second axis, respectively.
- For 3D models, select **1**, **2**, or **3** for the first, second, or third axis, respectively.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry Constraints

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Wire > **More Constraints** > **Antisymmetry**

Ribbon

Physics tab with **Wire** selected:

Points > **More Constraints** > **Antisymmetry**

Edge Load

Add an **Edge Load** as a force distributed along an edge (3D models) or boundary (2D models).

FORCE

Select a **Load type** — **Force per reference length**, **Force per deformed length**, or **Total force**. Enter values or expressions for the components.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per reference length	$\mathbf{f}_L$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Force per deformed length	$\mathbf{f}_l$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	edges boundaries	3D (x, y, z) 2D (x, y)

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the Wire interface with another physics interface, it is also possible to choose a predefined load from this list.

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

If you are performing a linear buckling analysis with a combination of live and dead loads, select the **Treat as dead load** checkbox to indicate that the load contributions from this node are constant. The default is that a load is proportional to the load factor.

For more information about live and dead loads, see Buckling Analysis.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Wire > **Line and Volume Loads** > **Edge Load**

Ribbon

Physics tab with **Wire** selected:

Edges > **Line and Volume Loads** > **Edge Load**

14

Pipe Mechanics

This chapter describes the Pipe Mechanics interface ( ), which you find under the **Structural Mechanics** branch ( ) when adding a physics interface.

The Pipe Mechanics Interface

The **Pipe mechanics (pipem)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is used for analysis of stresses and deformation in pipes. It can be modeled on 2D boundaries and 3D edges.

Two-noded straight elements with a Hermitian formulation are used. Two different assumptions about the physics can be used:

- Euler (or Euler-Bernoulli) theory. This formulation is intended for slender pipes, and do not take shear deformations into account.
- Timoshenko theory. In this formulation that extends the beam theory to 'thick' beams, shear deformations are taken into account. In a dynamic analysis, inertial effects from rotation are also included.

!	The formulation in this physics interface is valid only for straight pipes. In pipe bends, the ovalization of the cross section will cause a reduction in stiffness and increase in stresses. These effects are not accounted for.

Among the computed results are displacements, rotations, stresses, strains, and section forces. In addition to giving the pipe properties explicitly in terms of area, moment of inertia, and so on, predefined common cross-section types are available. Cross section data to be used in Pipe Cross Section settings can be computed using The Beam Cross Section Interface.

The material in the pipe is assumed to be linear elastic.

When this physics interface is added, these default nodes are also added to the Model Builder: Fluid and Pipe Materials, Pipe Cross Section, Fluid Load, Free (a condition where points are free, with no loads or constraints), and Initial Values. Then, from the **Physics** toolbar, add other nodes that implement, for example, loads and constraints. You can also right-click **Pipe Mechanics** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern

<name>.<variable_name>. In order to distinguish between variables belonging to different physics interfaces, the name string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is pipem.

BEAM FORMULATION

Select **Euler-Bernoulli** or **Timoshenko** to use the appropriate beam theory. Timoshenko theory, which is the default, includes the effects of shear flexibility and rotary inertia. Euler-Bernoulli theory is appropriate for pipes with cross section dimensions which are small relative to the length of the pipe, whereas Timoshenko theory can be used both for thick and slender pipes.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasistatic**. Use **Quasistatic** to treat the dynamic behavior as quasi static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasistatic** is selected.

This is often the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. Normally these settings do not need to be changed.

You can choose how extra ODE variables added by some features are grouped in the **Dependent Variables** node of a generated solver sequence.

- Select the **Rigid materials** checkbox to group variables added by Rigid Material nodes.
- Select the **Rigid connectors** checkbox to group variables added by Rigid Connector nodes.

The selection made in the **Advanced Settings** section can be overridden by the settings in the **Advanced** section of the Rigid Material or Rigid Connector features.

DISCRETIZATION

The discretization cannot be changed. The element has different shape functions for the axial and transversal degrees of freedom. The axial displacement and twist are represented by linear shape functions, while the bending is represented by a cubic shape function ("Hermitian element").

DEPENDENT VARIABLES

The Pipe Mechanics interface has these dependent variables (fields):

- The displacement field **u**, which has two components (`u`, `v`) in 2D and three components (`u`, `v`, and `w`) in 3D.
- The rotation angle θ, which has one component in 2D (`th`) and three components in 3D (`thx`, `thy`, and `thz`).

The names can be changed but the names of fields and dependent variables must be unique within a model.

The dependent variable names remain same in both a geometrically linear and a geometrically nonlinear analysis. Under geometric nonlinearity, the dependent variables are however not defined though shape functions. The equivalent shape function variables are (`pipem.uLinx`, `pipem.uLiny`, `pipem.uLinz`) and (`pipem.thLinx`, `pipem.thLiny`, `pipem.thLinz`). In this case, you will see the latter names under **Dependent Variables** in the **Solver Configurations** tree.

If needed, these shape variables can be used to write any extra contributions in the Pipe Mechanics interface.

If a physics interface that separates the material and spatial frame (Solid Mechanics is one such example) is added to the model, the coordinate indices change from (x, y, z) to (X, Y, Z) in the names of these variables.

- Boundary, Edge, Point, and Pair Nodes for the Pipe Mechanics Interface
- Theory for the Pipe Mechanics Interface

- *Coupled Analysis of Flow and Stress in a Pipe*: Application Library path **Structural_Mechanics_Module/Pipe_Mechanics/pipe_flow_stress**

Boundary, Edge, Point, and Pair Nodes for the Pipe Mechanics Interface

The Pipe Mechanics Interface has these boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Pipe Mechanics interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Edge** level (3D) or **Boundary** level (2D) are

- **Material Models**
- **Line and Volume Loads**
- **Mass, Spring, and Damper**
- **Line Constraints**

The submenus at the **Point** level are

- **Connections**
- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Bend
- Fluid and Pipe Materials
- Fluid Load
- Gravity
- Pipe Cross Section
- Rigid Material
- Section Orientation
- Thermal Expansion (for Materials)

These nodes are described for the Beam interface:

- Antisymmetry
- Edge Load
- External Stress
- Initial Stress and Strain
- Initial Values
- No Rotation
- Pinned
- Point Load
- Point Load, Free
- Point Mass
- Prescribed Acceleration
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Rigid Connector
- Symmetry
- Thermal Expansion (for Constraints)

These nodes are described for the Solid Mechanics interface:

- Added Mass
- Center of Rotation Nodes
- Damping
- Fixed Constraint
- Free
- Predeformation
- Safety
- Spring Foundation
- Spring–Damper

If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence. The exception is that the “Pinned” and “No Rotation” boundary conditions do not override each other since the degrees of freedom that they constrain are mutually exclusive.

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Pipe Cross Section

In the **Pipe Cross Section** node, you specify the geometric properties of the pipe’s cross section. In addition, some stress evaluation properties can be defined.

For 3D models, a default Section Orientation subnode is added, in which you specify the orientation of the principal axes of the section. You can add any number of **Section Orientation** subnodes if the same section appears with different spatial orientations in the structure.

	This is required input data.

	• Cross Sections • Beam Cross Section

PIPE SHAPE

The default is **Not Set**. Select **Circular**, **Square**, **Rectangular**, or **User defined**. Then go to the relevant section below to continue defining the section

For **User defined** go to Wall Cross Section Properties and Stress Evaluation Properties to continue defining the cross section.

	For equations and a figure see: • Circular Section • Square Section • Rectangular Section

Circular

Enter values or expressions for the following.

- **Outer diameter** d_o
- **Inner diameter** d_i

Piping standards may require stresses to be evaluated using a reduced thickness to account for possible corrosion or erosion effects on the pipe's wall. The thickness changes can occur on both the inner and the outer side of the wall. In the section *Thickness reduction for stress evaluation* enter values or expressions for the thickness reduction on the inner or outer sides:

- **Thickness reduction, outside** Δt_{o}
- **Thickness reduction, inside** Δt_{i}

For both sides the default value is `0[%]*pipem.t_nom`, where the variable `pipem.t_nom` is the nominal wall thickness given by d_o and d_i. The reduced inner and outer diameters are then defined as

$$d_{i,\,s} = d_i + 2\Delta t_i$$

$$d_{o,\,s} = d_o - 2\Delta t_o$$

Note that the reduced wall thickness only has an effect on the stress evaluation. That is, the flexibility of the piping system is still computed using nominal thicknesses. Internally, COMSOL Multiphysics creates separate variables for stiffness properties such as the cross-section area or the 2nd moment of inertia. All stress measures that can be evaluated in postprocessing are computed using the reduced wall diameters.

Square

Enter values or expressions for the following:

- **Inner width** h_i
- **Wall thickness** t

Rectangular

Enter values or expressions for the following:

- **Width in local y direction** h_i
- **Width in local z direction** w_i
- **Wall thickness in local y direction** t_y
- **Wall thickness in local z direction** t_z

WALL CROSS SECTION PROPERTIES

This section is only available if **User defined** is selected as the **Cross-Section Definition**.

The following table lists the basic section properties (some apply in 3D only). Enter values for these properties in the associated fields. The default values correspond to a circular cross section with a diameter of 0.1 m:

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Area of cross section	A	m^2
2D and 3D	Moment of inertia about local z-axis	I_{zz}	m^4

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D, Timoshenko beam	Shear correction factor along local y-axis	κ_y	1
3D only	Distance to shear center in local z direction	e_z	m
3D only	Moment of inertia about local y-axis	I_{yy}	m^4
3D only	Distance to shear center in local y direction	e_y	m
3D only	Torsional constant	J	m^4
3D only, Timoshenko beam	Shear correction factor along local z-axis	κ_z	1

For 3D models, the orientation of the cross section is given in Section Orientation. If the beam's cross section is a square or circle (solid or tube), the area moments of inertia are the same independent of direction, so the beam is totally symmetric and the orientation of the principal axes of the cross section is not a problem unless you are interested in looking at results defined using the local coordinate system. Such results are bending moments, shear forces, local displacements and rotations.

FLUID CROSS SECTION PROPERTIES

Enter the **Fluid cross section area**, A_f. This is the same as the internal cross section of the pipe.

STRESS EVALUATION PROPERTIES

This section is only available if **User defined** is selected as the **Cross-Section Definition**.

Select the **Bending stress evaluation points** — **From section heights** (the default) or **From specified points**.

Stress evaluation using only section heights is meaningful only when the cross section is symmetric.

The max shear stress factor determines the ratio between the peak and the average shear stress over the cross section as described by Equation 3-276 and Equation 3-277.

From Section Heights

For **From section heights** enter values in each field for the following parameters as needed for the space dimension:

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Section height in local y direction	h_y	m
2D and 3D	Max shear stress factor in local y direction	μ_y	1
3D only	Section height in local z direction	h_z	m
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	1
2D and 3D	Pressure membrane stress factor	γ_m	1
2D and 3D	Pressure bending stress factor	γ_b	1

From Specified Points

For **From specified points** enter values in the **Evaluation points in local system** table as needed for the space dimension. Then enter the following parameters in the applicable fields.

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Max shear stress factor in local y direction	μ_y	1
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	1
2D and 3D	Pressure membrane stress factor	γ_m	1
2D and 3D	Pressure bending stress factor	γ_b	1

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Pipe Cross Section**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Edges > **Pipe Mechanics** > **Pipe Cross Section**

Section Orientation

Use the **Section Orientation** subnode to define the orientation of a pipe cross section using a reference point or an orientation vector. There is always one **Section Orientation**

subnode for each cross section, and as many **Section Orientation** subnodes as needed can be added if the same section appears with different spatial orientations in the structure.

The most common pipe sections are symmetric. In this case, the only effect of the section orientation specification is to simplify the interpretation of section forces. This means that you in many cases can keep the default settings in this node.

This node is available for 3D components.

ORIENTATION METHOD

Select the **Reference point** (the default) or **Orientation vector**. For **Reference point**, enter a **Reference point defining local y direction** P.

The coordinate system is defined as follows:

The local x direction is in the edge direction. The positive edge direction can be checked by vector plotting the local edge tangent direction. The coordinates of the reference point define the local xy-plane together with the beam axis. The local coordinate system ($\mathbf{e}_{xl}$, $\mathbf{e}_{yl}$, $\mathbf{e}_{zl}$) is formed using the following algorithm:

$$\mathbf{v}_{zl} = \mathbf{e}_{xl} \times (\mathbf{p} - \mathbf{m})$$

$$\mathbf{e}_{zl} = \frac{\mathbf{v}_{zl}}{|\mathbf{v}_{zl}|}$$

$$\mathbf{e}_{yl} = \mathbf{e}_{zl} \times \mathbf{e}_{xl}$$

Here, **p** is the reference point, and **m** is the midpoint of the pipe element. The definition of the local coordinate system is illustrated in Figure 14-1.

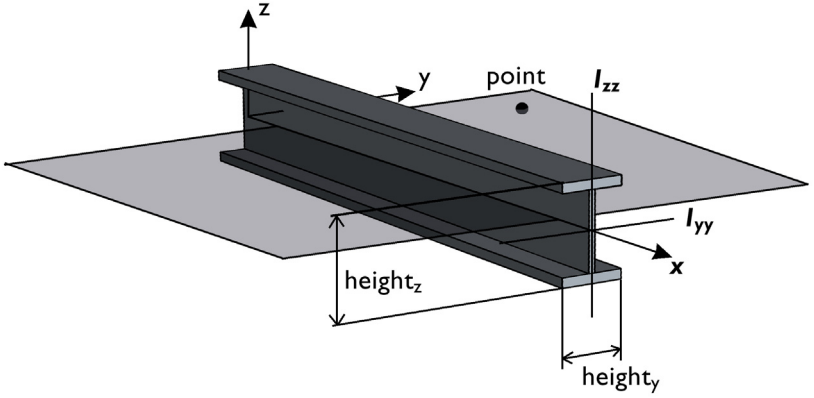


Figure 14-1: Local beam coordinate system defined by a reference point.

For the creation of a local coordinate system to be possible, the point cannot coincide with the edge or the edge extension. If this is attempted, an error message is generated.

The settings for the global coordinates of the point are `[1000,1000,1000]`. This is useful only for symmetric cross sections.

Often a number of edges in a plane have the same orientation. It is then easy to select all edges and specify a point anywhere in the same plane, not coinciding with an edge or an edge extension.

For **Orientation vector** enter **Orientation vector defining local y direction**, $\mathbf{V}$, and optionally the **Rotation of vector around beam axis** ϕ. The beam orientation is defined similarly to what is described above, with the difference that in this case the direction vector is explicitly defined whereas when an orientation point is used, the direction vector is obtained as the vector from the beam axis to the specified point. The local coordinate system ($\mathbf{e}_{xl}$, $\mathbf{e}_{yl}$, $\mathbf{e}_{zl}$) is formed using the following algorithm:

$$\mathbf{v}_{zl} = \mathbf{e}_{xl} \times \mathbf{V}$$

$$\mathbf{e}_{zl} = \frac{\mathbf{v}_{zl}}{|\mathbf{v}_{zl}|}$$

$$\mathbf{e}_{yl} = \mathbf{e}_{zl} \times \mathbf{e}_{xl}$$

The **Rotation of vector around beam axis** has the effect of rotating the given vector around the beam axis (using the right-hand rule) before it is used to define the local *xy*-plane. This simplifies the input for some cross sections, such as L-shaped profiles, where the principal axes have a direction which is skewed relative to a more natural modeling position. This can be written as

$$\mathbf{e}_{y1} = \mathbf{e}'_{y1}\cos\phi - \mathbf{e}'_{z1}\sin\phi$$

$$\mathbf{e}_{z1} = \mathbf{e}'_{z1}\cos\phi + \mathbf{e}'_{y1}\sin\phi$$

Here the directions denoted with a prime are unrotated beam axis orientations obtained by the procedure described above.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Pipe Cross Section** > **Section Orientation**

Ribbon

Physics tab with **Pipe Cross Section** node selected in the model tree:

Attributes > **Section Orientation**

Bend

Use the **Bend** subnode to account for a local stiffness reduction and a local stress increase in a bent pipe. Curved pipe segments with a circular cross section tend to ovalize when loaded in bending. This three-dimensional effect cannot be resolved with classical beam theory, but it may have a significant effect on flexibility and stress results. It is therefore common practice to reduce the pipe stiffness in a bend with so-called *flexibility factors*. Similarly, to account for the stress increase due to shape changes of the cross section so-called *stress intensification factors* (SIFs) are applied. Both flexibility factors and SIFs often depend on geometric properties of the pipe such as the bend radius, mean wall radius, and the wall thickness. Sometimes also the internal pressure is accounted for, which tends to counteract the ovalization.

The **Bend** feature assumes that the bend is modeled with a circular arc. The local y direction of the bend points in the direction of the curvature vector; that is, bending about the y-axis is defined as out-of-plane bending, and bending about the z-axis is

defined as in-plane bending. Since **Bend** defines its own coordinate system, it overrides the **Section Orientation** node.

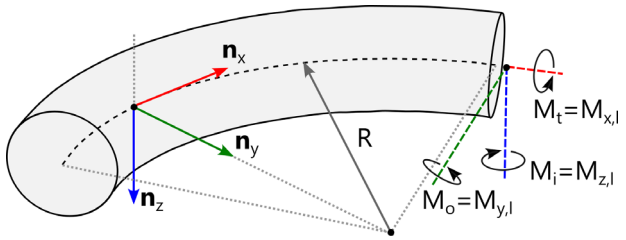


Figure 14-2: Local beam coordinate system for a bend.

In the **Bend** feature enter one or more of the following correction factors:

- **Flexibility factor, in-plane bending** k_{i}
- **Flexibility factor, out-of-plane bending** k_{o}
- **Flexibility factor, torsion** k_{t}
- **Flexibility factor, axial load** k_{a}
- **Stress intensification factor, in-plane bending** i_{i}
- **Stress intensification factor, out-of-plane bending** i_{o}
- **Stress intensification factor, torsion** i_{t}
- **Stress intensification factor, axial load** i_{a}

All flexibility factors and SIFs are by default equal to 1, that is, they have no effect on the calculation of displacements or stresses. All factors must by definition be greater or equal to 1. The truncation at the lower limit is automatically taken care of.

The flexibility factors do not directly reduce the stiffness properties, such as the moment of inertia I_{yy} and I_{zz}, as this would affect the stress evaluation as well. Instead, the stiffness reduction is effectively performed when assembling the global stiffness matrix, where the element matrices of bent sections use reduced stiffness properties.

Since the application of SIFs often differs between different standards, SIFs are applied only to a set of special stress quantities available for postprocessing. They are:

Maximum in-plane bending stress: `pipem.sbi_SIF`.

Maximum out-of-plane bending stress: `pipem.sbo_SIF`.

Axial stress at centerline: `pipem.sa_SIF`.

Maximum torsional shear stress: `pipem.ttmax_SIF`.

Other stress quantities must be modified manually if needed.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Pipe Cross Section** > **Bend**

Ribbon

Physics tab with **Pipe Cross Section** node selected in the model tree:

Attributes > **Bend**

Fluid and Pipe Materials

The **Fluid and Pipe Materials** node is the main node in the Pipe Mechanics interface. This is where you enter the material data for both the pipe itself and the fluid contained in the pipe.

It is common that the material data for the fluid and the pipe material are in located in different **Material** nodes. Only one material can however be assigned to a certain edge. For this reason, the assignment of material data can be made not only using the material assigned to the edge, but by accessing any **Material** node present in the component.

By adding the following subnodes to the **Fluid and Pipe Materials** node you can incorporate many other effects:

- Thermal Expansion (for Materials)
- Initial Stress and Strain
- External Stress

- Safety
- Damping

FLUID PROPERTIES

For the structural analysis point of view, the only required property of the fluid is its mass density.

For input of the **Density**, ρ_f, select either **From material** or **User defined**. When using **From material**, also select the material to use in the **Fluid material** list. This list contains all materials present in the component, as well as the default choice **Edge material**. When **Edge material** is selected, the material currently assigned to the edge (2D: boundary) is used.

For **User defined**, enter a value or expression.

PIPE PROPERTIES

In this section, you define the linear elastic material properties.

If the option From material used for any of the material properties below, you must select a material from the **Pipe material** list. This list contains all materials present in the component, as well as the default choice **Edge material**. When **Edge material** is selected, the material currently assigned to the edge (2D: boundary) is used.

Specification of Elastic Properties for Isotropic Materials

From the **Specify** list, select a pair of elastic properties for an isotropic material. Select:

- **Young's modulus and Poisson's ratio** to specify Young's modulus (elastic modulus) E and Poisson's ratio ν.
- **Young's modulus and shear modulus** to specify Young's modulus (elastic modulus) E and the shear modulus G.
- **Bulk modulus and shear modulus** to specify the bulk modulus K and the shear modulus G.
- **Lamé parameters** to specify the Lamé parameters λ and μ.
- **Pressure-wave and shear-wave speeds** to specify the pressure-wave speed (longitudinal wave speed) c_p and the shear-wave speed (transverse wave speed) c_s.

This is the wave speed for a solid continuum. In a pipe element, the actual speed with which a longitudinal wave travels is lower than the value given. When using this type of input the density must also be given.

For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Each of these pairs define the elastic properties, and it is possible to convert from one set of properties to another.

Density

Define the **Density** ρ of the material. Select **From material** to take the value from the material or **User defined** to enter a value for the density.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large rotation formulation for all edges. Select the **Geometrically linear formulation** checkbox to always use a small rotation formulation for the edges that have this material assigned, irrespective of the setting in the study step.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Material Models** > **Fluid and Pipe Materials**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Edges > **Material Models** > **Fluid and Pipe Materials** (3D)
Boundaries > **Material Models** > **Fluid and Pipe Materials** (2D)

Rigid Material

Add the **Rigid Material** node and select one or more edges (2D: boundaries) to make them a rigid body. **Rigid Material** is a material model, with only two material properties: the mass densities of the pipe and the fluid.

It is common that the material data for the fluid and the pipe material are in located in different **Material** nodes. Only one material can however be assigned to a certain edge. For this reason, the assignment of material data can be made not only using the

material assigned to the edge, but by accessing any **Material** node present in the component.

By default, an **Initial Values** node is added (see Initial Values (Rigid Material)).

You can add functionality to the rigid domain through the following subnodes:

- Fixed Constraint (Rigid Material) to fully constrain the rigid domain.
- Prescribed Displacement/Rotation to prescribe the displacement of individual degrees of freedom.
- Applied Force (Rigid Material) to apply a force in given point.
- Applied Moment (Rigid Material) to apply a moment.
- Mass and Moment of Inertia (Rigid Material) to add extra mass and moment of inertia in a given point.
- Spring Foundation (Rigid Material) to add a translational or rotational spring or damper in a given point.

FLUID PROPERTIES

In this section, you enter the density of the fluid.

For input of the **Density**, ρ_f, select either **From material** or **User defined**. When using **From material**, also select the material to use in the **Fluid material** list. This list contains all materials present in the component, as well as the default choice **Edge material**. When **Edge material** is selected, the material currently assigned to the edge (2D: boundary) is used.

For **User defined**, enter a value or expression.

PIPE PROPERTIES

In this section, you enter the density of the pipe material.

For input of the **Density**, ρ, select either **From material** or **User defined**. When using **From material**, also select the material to use in the **Fluid material** list. This list contains all materials present in the component, as well as the default choice **Edge material**. When **Edge material** is selected, the material currently assigned to the edge (2D: boundary) is used.

For **User defined**, enter a value or expression.

All other settings for the **Rigid Material** node are described in the documentation for Rigid Material in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Material Models** > **Rigid Material**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Edges > **Material Models** > **Rigid Material** (3D)

Boundaries > **Material Models** > **Rigid Material** (2D)

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature. The thermal strain depends on the coefficient of thermal expansion (CTE) α, the temperature T, and the strain-free reference temperature T_{ref} as

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

The average increase in temperature will cause a global extension of the pipe, whereas a difference between the internal and external temperatures will cause local bending stresses.

The effects of a thermal gradient through the pipe wall is only taken into account for the Circular Section.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

The **Temperature**, T, input is not used. All temperature information is taken from the inputs in the **Thermal Expansion Properties** section. This field will, for formal reasons,

be visible in the GUI indicating when the selected material has a temperature dependent coefficient of thermal expansion.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain.

From the **Coefficient of thermal expansion** α list, select **From material** to use the coefficient of thermal expansion from a material, or **User defined** to enter a value or expression for α. When **From material** is used, then you need to specify which material to use by selecting a **Pipe material**. This list contains all materials present in the component, as well as the default choice **Edge material**. When **Edge material** is selected, the material currently assigned to the edge (2D: boundary) is used.

Enter the **Inside temperature**, T_{in}, and **Outside temperature**, T_{out}.

If the temperature is computed in the Heat Transfer in Pipes interface, there are two fundamentally different cases:

- The temperature gradient in the thickness direction is ignored. In this case you can use the temperature of the fluid as both inner and outer temperature, assuming that the temperature of the pipe is the same as that of the fluid. Do this by setting the **Temperature** to that of the Heat Transfer in Pipes interface, typically **Temperature (htp)**.
- The temperature difference in the thickness direction is computed, using **Wall Heat Transfer** in the Heat Transfer in Pipes interface. In this case, use the **Temperature** list's **User defined** option and enter the variables defined in the **Wall Layer** node the for the inner and outer temperatures. These variables typically have names like `htp.wht1.wall1.Tin` and `htp.wht1.wall1.Tout`.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Fluid and Pipe Materials** > **Thermal Expansion**

Ribbon

Physics tab with **Fluid and Pipe Materials** node selected in the model tree:

Attributes > **Thermal Expansion**

Gravity

When you add a **Gravity** node, gravity forces are applied to features in the physics interface with a density, mass, or mass distribution.

The fluid in the pipe is included in the direction transverse to the pipe, but not in the axial direction. To include the gravity load in the axial direction caused by mass of the fluid, you will typically have to add extra point loads at discontinuities like pipe bends.

The gravity acts in a fixed spatial direction. The load intensity is $\mathbf{g} = \rho g \mathbf{e}_g$ where g as a default is the acceleration of gravity (a predefined physical constant).

COORDINATE SYSTEM SELECTION

Select a coordinate system in which the gravity field vector is represented. Only systems with space-independent axis directions are consistent with the assumptions of this feature. The **Global coordinate system** is selected by default.

GRAVITY

Enter the components for **Gravity g**. The default value is `g_const`, which is the gravitational acceleration constant at mean sea level having the value 9.80665 $\mathrm{m/s^2}$.

For 2D components, the default is that the gravity acts in the negative y direction. For 3D components, the default is that the gravity acts in the negative z direction.

!	Only features that have a geometrical selection contribute to the mass forces. The **Mass and Moment of Inertia** nodes are global features and will not get any contribution from **Gravity**. You may need to add extra force contributions if such nodes are present.

	Frame Acceleration Forces

	In versions earlier than 6.1, the **Gravity** node was a feature with domain selection. In a model created in an earlier version, the old **Gravity** node remains unchanged. For such a model, it is recommended to use the newer functionality. There are different options: • In most cases, switching to the new version of **Gravity** is appropriate. • If the current **Gravity** node has a selection, which is not the entire model, or if it contains expressions that depends on the spatial coordinates, you can replace it by a Linearly Accelerated Frame node. • If the current **Gravity** node is used to represent an acceleration imposed by the foundation, consider replacing it by a Base Excitation node.

	• You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type. • You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Gravity**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Global > **Gravity**

Base Excitation

The **Base Excitation** node is used to represent an acceleration that is applied at all constraints. The main use case is for modal based analyses, in which case it is not possible to directly prescribe nonzero displacements at constrained degrees of freedom. It is however possible to use **Base Excitation** in any type of dynamic analysis.

The fluid in the pipe is included in the direction transverse to the pipe, but not in the axial direction. To include the inertial load in the axial direction caused by mass of the fluid, you will typically have to add extra point loads at discontinuities like pipe bends.

When a structure is subjected to a base excitation, displacements, velocities, and accelerations are measured relative to a space fixed coordinate system.

COORDINATE SYSTEM SELECTION

Select a coordinate system in which the base acceleration vector is represented. Only systems with space-independent axis directions are consistent with the assumptions of the base excitation.

BASE EXCITATION

Enter the **Base acceleration,** $\mathbf{a}_b$.

	A degree of freedom should either be constrained or free in the direction where the base excitation acts. Having a constraint acting in a local direction that is not either aligned or orthogonal to the excitation may produce unexpected results.

	Only features that have a geometrical selection contribute to the mass forces. The **Mass and Moment of Inertia** nodes are global features and will not get any contribution from **Base Excitation**. You may need to add extra force contributions if such nodes are present.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Base Excitation**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Global > **Base Excitation**

Linearly Accelerated Frame

When you add a **Linearly Accelerated Frame** node, inertial forces are applied to all selected features in the physics interface with a density, mass, or mass distribution. You

select objects having the highest geometrical dimension of the interface, and all objects with a lower dimensionality that belong to the selection are automatically included.

The fluid in the pipe is included in the direction transverse to the pipe, but not in the axial direction. To include the inertial load in the axial direction caused by mass of the fluid, you will typically have to add extra point loads at discontinuities like pipe bends.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. It can be used when prescribing the direction of the frame acceleration.

LINEARLY ACCELERATED FRAME

Enter the components of the linear **Frame acceleration**, $\mathbf{a}_f$. The force is computed as $\mathbf{f} = -\rho\mathbf{a}_f$ and acts in a fixed spatial direction.

TIME-DEPENDENT STUDY

For a time-dependent study, you can compute the absolute velocity and absolute displacement postprocessing variables by checking the corresponding checkboxes.

Frame Acceleration Forces

Only features that have a geometrical selection contribute to the mass forces. The **Mass and Moment of Inertia** nodes are global features and will not get any contribution from **Linearly Accelerated Frame**. You may need to add extra force contributions if such nodes are present.

- You can add a Harmonic Perturbation subnode for specifying a harmonic variation of the values of the prescribed acceleration.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Line and Volume Loads** > **Linearly Accelerated Frame**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Edges > Line and Volume Loads > Linearly Accelerated Frame

Fluid Load

Add a **Fluid Load** node to describe loads from the fluid on the pipe. Three types of loads can be entered: Internal pressure, drag force from the fluid, and centrifugal forces in pipe bends.

When computing the effect of the pressure loads, only the internal gauge pressure is important, as discussed in Effects of Internal Pressure. You can thus use two approaches:

- Enter the gauge pressure as internal pressure, and set the external pressure to zero. This is a common approach when the Pipe Mechanics interface is used standalone.
- Enter absolute pressures for both the internal and external pressures. When combining with the Pipe Flow interface this is the common case, since the pressure in that interface is often interpreted as being absolute.

Since pressure loads are almost invariably part of a piping analysis, one **Fluid Load** node with all edges selected is included as default when the physics interface is created. It is however not mandatory to use it, and it will not give any contributions to the solution unless you have entered data in it. You can add any number of **Fluid Load** nodes.

FLUID LOADS

Enter the **Internal pressure**, p_i. The list contains any pressure variables announced by other physics interfaces, as well as the option **User defined**. For **User defined**, enter a value or an expression for the internal pressure.

Select a method for entering the **External pressure**, p_o, — **User defined** or **Common model input**.

For **User defined**, enter a value or an expression for the external pressure.

Select a method for entering the **Drag force per reference length**, F_d. The choice **User defined** is always available, in which case you enter a value or expression in the text field. A positive load acts in the same direction as the positive direction of the edge to which

the load is applied. To visualize edge orientations, select **Show edge direction arrows** in the appropriate **View** node.

The **Drag force per reference length** list can also contain drag force variables announced by the Pipe Flow interface.

Select a method for entering the **Centrifugal force per reference length**, F_c. The choice **User defined** is always available, in which case you enter values or expressions in the text fields. In the **User defined** case, the input is similar to using an Edge Load.

The **Centrifugal force per reference length** list can also contain drag force variables announced by the Pipe Flow interface.

See also

- Effects of Internal Pressure and Fluid Flow Forces in the Theory for the Pipe Mechanics Interface section
- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*.

You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Pipe Mechanics > **Line and Volume Loads** > **Edge Load**

Ribbon

Physics tab with **Pipe Mechanics** selected:

Edges > **Line and Volume Loads** > **Fluid Load** (3D)
Boundaries > **Line and Volume Loads** > **Fluid Load** (2D)

15

Phase Field in Solids

This chapter describes the Phase Field in Solids interface ( ), which is found in the **Moving Interfaces** folder ( ) under the **Mathematics** branch ( ) when adding a physics interface.

The Phase Field in Solids Interface

The **Phase Field in Solids (pfs)** interface ( ), found in the **Moving Interfaces** folder ( ) under the **Mathematics** branch ( ) when adding a physics interface, is used for modeling the evolution of sharp interfaces in solids, such as those associated with damage, fracture, and grain boundaries.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

When this physics interface is added, these default nodes are also added to the Model Builder: **Phase Field Model**, **No Flux**, and **Initial Values**. Then, from the **Physics** toolbar, you can add other nodes that implement, for example, boundary conditions and source terms. You can also right-click **Phase Field in Solids** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `pfs`.

DISCRETIZATION

In the Phase Field in Solids interface, you can choose not only the order of the discretization, but also the type of shape functions: *Lagrange* or *serendipity*. For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will however give significant reductions of the model size for a given mesh containing hexahedral, prism, or quadrilateral elements. In 1D components, there is no difference between Lagrange and serendipity shape functions.

The default is to use **Linear** shape functions.

To display other settings for this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

See Table 2-4 for links to common sections and Table 2-5 for common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

DEPENDENT VARIABLES

Define the number of the dependent variables and the variable names. The default name for a phase field is *phi*.

The dependent variables and their names must be unique with respect to all other dependent variables in the model.

Enter the **Number of phase field variables**. Use the **Add Phase Field** ( ) and **Remove Phase Field** ( ) buttons as needed.

- Domain and Boundary Nodes for the Phase Field in Solids Interface
- Theory for the Phase Field in Solids Interface

Domain and Boundary Nodes for the Phase Field in Solids Interface

The Phase Field in Solids Interface has these domain, boundary, and pair nodes, which are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- Bounds
- Initial Values
- No Flux
- Phase Field Model
- Prescribed Phase Field
- Source
- Symmetry

In the *COMSOL Multiphysics Reference Manual* see Table 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Phase Field Model

The **Phase Field Model** node adds the equations to model phase field transport in solids. It allows for specifying a scalar potential function of all phase field variables as well as the length scale, structure tensor, and regularization time of each phase field variable.

PHASE FIELD MODEL

Select a **Phase field model** — **Quadratic**, **AT1 (Ambrosio–Tortorelli type 1)**, **AT2 (Ambrosio–Tortorelli type 11)**, **PF–CZM (Phase-field regularized cohesive zone model)**, or **User defined**. Here, **AT1**, **AT2**, and **PF–CZM** are specialized models for modeling phase-field damage and fracture in solids; these models are only available when the interface is used with a single phase field variable.

For the **User defined** formulation, enter an expression for the potential function as a function of the dependent variables (phase field variables).

Enter a **Length scale**, l_{int}, for each of the phase field variables.

If the **Quadratic** or **User defined** formulation is chosen, enter the **Structure tensor**, D, for each of the phase field variables. Select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components of the structure tensor. When a **Diagonal** or **Symmetric** tensor input is used, the axis orientations are given by the coordinate system selection.

Enter a **Viscous regularization time**, τ, for each of the phase field variables.

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **Phase Field Model**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Domains > **Phase Field Model**

Bounds

The **Bounds** node adds inequality constraints on the phase field variables, which can be useful to limit the values of the dependent variables.

BOUNDS

Enter a value for the upper and/or lower bound for each of the phase field variables in the table as needed. Select the corresponding checkboxes to activate the bounds.

CONSTRAINT FORCE

Enter a **Penalty factor**, K. A larger penalty factor reduces the violation of the constraint, but a value that is too large can result in an ill-conditioned stiffness matrix.

By default, the constraint force associated with an active bound is computed as the penalty factor K multiplied by the violation of the constraint. For example, the constraint force for a lower bound $\mathrm{lb} < \phi$ is computed as $f_c = K\max(-\phi + \mathrm{lb}, 0)$. In some cases, the discontinuous derivative of the `max()` operator may cause convergence issues. It is possible to use a smooth version of the constraint force by selecting the **Use transition zone** checkbox. When selected, enter the size of the transition zone, ϕ_{tr}. This positive number should be set based on the constraint violation that is deemed acceptable. Both the sharp and the smooth constraint force do not add a penalty contribution as long as the inequality is satisfied. The penalty factor K will be applied in the constraint force expression for all active bounds.

LOCATION IN USER INTERFACE

Context Menus

Phase Field Model > **Bounds**

Ribbon

Physics tab with **Phase Field Model** selected:

Attributes > **Bounds**

Prescribed Phase Field

The **Prescribed Phase Field** node adds a condition where the phase field variables are prescribed in the geometric entity selected (domain, boundary, edge, or point).

PRESCRIBED PHASE FIELD

If there is only one phase field variable, enter its prescribed value, ϕ_0.

If there are two or more phase field variables, select the corresponding checkbox for those phase fields to be prescribed and enter their values $\phi_{0,k}$.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **Prescribed Phase Field**
Phase Field in Solids > **Edges** > **Prescribed Phase Field**
Phase Field in Solids > **Points** > **Prescribed Phase Field**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Domains > **Prescribed Phase Field**
Boundaries > **Prescribed Phase Field**
Edges > **Prescribed Phase Field**
Points > **Prescribed Phase Field**

Source

The **Source** node describes the driving force for the evolution of the phase field within the domain. Add one or more nodes as needed — all sources within a domain contribute with each other.

SOURCE TERM

Enter the value or expression of the source term, f, for each of the phase field variables that should contribute to the right-hand side of the phase field equation.

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **Source**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Domains > **Source**

Initial Values

The **Initial Values** node specifies the initial values for each phase field variable. These serve as an initial guess for a stationary solver or as initial conditions for a transient simulation.

INITIAL VALUES

Enter a value or expression for the initial value of the dependent variables.

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **Initial Values**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Domains > **Initial Values**

No Flux

The **No Flux** node is the default boundary condition on exterior boundaries. It should be used on boundaries across which there is no flux. The condition applied for each phase field variable ϕ_k corresponds to

$$-\mathrm{D}_k \nabla \phi_k \cdot \mathbf{N} = 0$$

where D_k denotes the structure tensor and $\mathbf{N}$ denotes the outward pointing normal of the boundary in the material frame.

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **No Flux**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Boundaries > **No Flux**

Symmetry

The **Symmetry** node can be used to represent boundaries where the phase field variable distribution is symmetric, that is, where there is no flux across the boundary. This boundary condition is identical to that of the No Flux node. It can be applied on exterior boundaries only.

LOCATION IN USER INTERFACE

Context Menus

Phase Field in Solids > **Symmetry**

Ribbon

Physics tab with **Phase Field in Solids** selected:

Boundaries > **Symmetry**

16

Transport in Solids

This chapter describes the Transport in Solids interface ( ), which you find under the **Chemical Species Transport** branch ( ) when adding a physics interface.

The Transport in Solids Interface

The **Transport in Solids (ts)** interface ( ), found under the **Chemical Species Transport** branch ( ), is used to model diffusion phenomena in a solid.

For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

The interface supports simulation of transport by diffusion in 1D, 2D, and 3D as well as in axisymmetric components in 1D and 2D. The default unit of the dependent variable is the molar concentration, but other units can be used, such as mass density or number density. Modeling multiple species transport is possible, whereby the physics interface solves for the selected quantity of each species.

When this physics interface is added, these default nodes are also added to the Model Builder: **Solid**, **No Flux** (prescribing no flux in the diffused variable across all exterior boundaries), and **Initial Values**. Then, from the **Physics** toolbar, you can add other nodes that implement, for example, boundary conditions and source terms. You can also right-click **Transport in Solids** to select physics nodes from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `ts`.

DOMAIN SELECTION

If any part of the model geometry should not partake in the diffusion model, remove that part from the selection list.

OUT-OF-PLANE THICKNESS

For 2D components, the thickness field (default value: 1 m) defines a parameter for the thickness of the geometry perpendicular to the two-dimensional cross-section.

CROSS-SECTIONAL AREA

For 1D components, enter a cross-sectional area A_c (SI unit: m^2) to define a parameter for the area of the geometry perpendicular to the 1D component. The default value is 1 m^2.

TRANSPORTED QUANTITY

From the **Transported quantity** list, select **Concentration**, **Mass density**, or **Number density** to assign the unit for the dependent variable.

DISCRETIZATION

In the Transport in Solids interface, you can choose not only the order of the discretization, but also the type of shape functions: *Lagrange* or *serendipity*. For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will however give significant reductions of the model size for a given mesh containing hexahedral, prism, or quadrilateral elements. In 1D components there is no difference between Lagrange and serendipity shape functions. The default is to use **Quadratic Lagrange** shape functions.

To display other settings for this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

The **Compute boundary fluxes** checkbox is activated by default so that COMSOL Multiphysics computes predefined accurate boundary flux variables. When this option is selected, the solver computes variables storing accurate boundary fluxes from each boundary into the adjacent domain.

The flux variable affected in the interface is `ntflux_c`, where `c` is the dependent variable. This is the normal total flux and corresponds to the boundary flux plus additional external flux.

Also the **Apply smoothing to boundary fluxes** checkbox is available if the previous checkbox is selected. The smoothing can provide a more well-behaved flux value close to singularities.

For details about the boundary fluxes settings, see Computing Accurate Fluxes in the *COMSOL Multiphysics Reference Manual*.

The **Value type when using splitting of complex variables** setting should be set to Real, which is the default.

DEPENDENT VARIABLES

Define the number of the dependent variables and the variable names. The default name for the transported quantity field is *c* by default.

	The dependent variables and their names must be unique with respect to all other dependent variables in the model.

Add or remove dependent variables in the model and also change their names.

Enter the **Number of dependent variables**. Use the **Add transported quantity** (+) and **Remove transported quantity** ( ) buttons as needed.

	• Domain and Boundary Nodes for the Transport in Solids Interface • Theory for the Transport in Solids Interface

	Hydrogen Diffusion in Metals: Application Library path **Structural_Mechanics_Module/Diffusion_in_Solids/hydrogen_diffusion**

Domain and Boundary Nodes for the Transport in Solids Interface

The Transport in Solids Interface has these domain, boundary, and pair nodes, which are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

	In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section:

- Chemical Potential
- External Chemical Potential
- External Flux
- Electromigration
- Stress Migration
- Thermophoresis
- Electrode Surface Coupling
- Flux
- Initial Values
- No Flux
- Prescribed Chemical Potential
- Prescribed Transported Quantity
- Periodic Condition
- Solid
- Source
- Symmetry

You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Chemical Potential

The **Chemical Potential** node adds the equations to model diffusive flux in terms of the gradient of the chemical potential. This node is active only when **Concentration** is selected as **Transported Quantity**.

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (-M_i \nabla \mu_i) = S_i$$

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame.

MOBILITY

Select a **Mobility model** — **Linear**, **Quadratic**, or **User defined**.

For **Linear**, enter expressions for the **Mobility**, η_c, (s mol/kg) for each of the species.

If **Quadratic** is selected as the **Mobility model**, enter also the **Maximum concentration** $c_{\max}$ for each of the species. The default value is `Inf` mol/m^3.

For **User defined**, enter expressions for the **Mobility coefficient**, M_c, for each of the species. Select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components. When an anisotropic input is used, the axis orientations are given by the Coordinate System Selection.

CHEMICALLY STORED FREE ENERGY

Under **Chemically stored free energy model**, select **User defined** or **Flory–Huggins** when there is only one species. For **User defined**, enter expressions for the **Molar volume**, $V_{m,c}$, one expression per species. For **Flory–Huggins**, enter expressions for the **Interaction parameter**, χ_c, the **Molar volume**, $V_{m,c}$, and the **Stretch**, λ_0.

- The **Interaction parameter**, χ_c, is a dimensionless measure of the enthalpy of mixing between the solvent and polymer network. An expression can be inserted to simulate dependence from other parameters such as temperature.
- The **Stretch**, λ_0, or free swelling stretch, is used to take into account that the meshed geometry is a result of an unconstrained expansion (free swelling) from a dry configuration (zero species concentration). This is necessary because the Flory–Huggins model is singular at zero concentration, so an initial condition different from zero should be used in the Initial Values node. The stretch value should be compatible with the selected initial conditions.

For details, see the Chemical Potential section in the *Structural Mechanics Theory* chapter.

If multiple species are present, only the **User defined** option is available as the **Chemically stored free energy model**.

ADVANCED

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the mobility tensor in the material frame or in the spatial frame. They are related as follows,

$$M_i = JF^{-1}m_iF^{-T}$$

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **Chemical Potential**

Ribbon

Physics tab with **Transport in Solids** selected:

Domain > **Chemical Potential**

External Chemical Potential

The **External Chemical Potential** subnode is used to add an additional contribution to the total chemical potential coming from free energy contributions not included in the Chemical Potential parent node.

$$\mu_i = \mu_i + \mu_{\text{ext},i}$$

The additional chemical potential contributes to the total flux when it has a spatial variation.

$$\Gamma_i = \Gamma_i - M_i \nabla \mu_{\text{ext},i}$$

EXTERNAL CHEMICAL POTENTIAL

Enter a value or expression in the corresponding field for each species.

LOCATION IN USER INTERFACE

Context Menus

Chemical Potential > **External Chemical Potential**

Ribbon
Physics tab with **Chemical Potential** selected:

Attributes > **External Chemical Potential**

External Flux

The **External Flux** subnode is used to add an additional flux contribution ($\Gamma_{\text{ext},\,i}$) which is not part of the constitutive relation. Multiple subnodes can be added. The mass conservation is modified as follows,

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (\Gamma_i + \Gamma_{\text{ext},\,i}) = S_i$$

EXTERNAL FLUX

Enter a value or expression for the additional flux in the corresponding field.

ADVANCED

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the flux in the reference (material frame) or in the deformed (spatial frame) configuration.

LOCATION IN USER INTERFACE

Context Menus
Solid > **External Flux**

Ribbon
Physics tab with **Solid** selected:

Attributes > **External Flux**

Electromigration

The **Electromigration** subnode is used to add an additional flux contribution ($\Gamma_{\text{em},\,i}$) that is proportional to the gradient of an electric field. The flux from electromigration is evaluated as follows

$$\Gamma_{\text{em},\,i} = -Mz_iF\nabla V$$

where z_i are the charge number of each species, F is Faraday constant, and V is the electric potential. The flux is evaluated in the reference or deformed configuration depending on the choice for **Frame** in the **Solid** node.

ELECTROMIGRATION

The **Electric Potential** can be either a **User defined** expression or it can be selected from another physics interface.

Specify the **Charge number** for each species, and then enter a value or expression in the corresponding field.

See also Electromigration in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid > **Electromigration**

Ribbon

Physics tab with **Solid** selected:

Attributes > **Electromigration**

Stress Migration

The **Stress Migration** subnode is used to add an additional flux contribution ($\Gamma_{\mathrm{sm},\,i}$) proportional to the gradient of the mean stress. The flux from stress migration is evaluated as follows

$$\Gamma_{\mathrm{sm},\,i} = M_i \nabla(V_{\mathrm{m}i}\sigma_{\mathrm{m}})$$

in which M_i is the mobility as selected in the **Solid** node, $V_{\mathrm{m}i}$ is the molar volume of each species, and σ_{m} is the mean stress. The flux is evaluated in the reference or in the deformed configuration depending on the choice for **Frame** in the **Solid** node.

STRESS MIGRATION

The **Mean stress** can be either a **User defined** expression or it can be selected from another physics interface.

Enter the **Molar volume** of each species.

See also Stress Migration in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus
Solid > **Stress Migration**

Ribbon
Physics tab with **Solid** selected:

Attributes > **Stress Migration**

Thermophoresis

The **Thermophoresis** subnode is used to add an additional flux contribution ($\Gamma_{\mathrm{em},\,i}$) proportional to the gradient of the temperature. The flux from thermophoresis is evaluated as follows

$$\Gamma_{\mathrm{th},\,i} = -M_i S_{\mathrm{T}i} R \nabla T$$

in which M_i is the mobility, and $S_{\mathrm{T}i}$ is Soret coefficient of each species, R is the universal gas constant, and T is the temperature. The mobility and the temperature are used as specified in the **Solid** node. The flux is evaluated in the reference or in the deformed configuration depending on the choice for **Frame** in the **Solid** node.

THERMOPHORESIS

Specify if the flux from thermophoresis is evaluated using the **Soret coefficient** or the **Thermal diffusion coefficient**. The thermal diffusion coefficient is only available if the **Mobility model** is not set to **User defined** in the **Solid** node. The thermal diffusion coefficient and the diffusion coefficient are related by the Soret coefficient

$$D_{\mathrm{T}i} = S_{\mathrm{T}i} D_i$$

See also Thermophoresis in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid > **Thermophoresis**

Ribbon

Physics tab with **Solid** selected:

Attributes > **Thermophoresis**

Initial Values

The **Initial Values** node specifies the initial values for the dependent variable of each species. These serve as an initial guess for a stationary solver or as initial conditions for a transient simulation.

INITIAL VALUES

Enter a value or expression for the initial value of the dependent variable selected (concentration, mass density, or number density). This also serves as an initial guess for stationary problems. The SI unit depends on the selection done in Transported Quantity.

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **Initial Values**

Ribbon

Physics tab with **Transport in Solids** selected:

Domains > **Initial Values**

No Flux

The **No flux** node is the default boundary condition on exterior boundaries. It should be used on boundaries across which there is no flux. The condition applied for each species corresponds to

$$\mathbf{N} \cdot \Gamma_i = 0$$

where $\mathbf{N}$ denotes the outward pointing normal of the boundary, in the reference configuration.

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **No Flux**

Ribbon

Physics tab with **Transport in Solids** selected:

Boundaries > **No Flux**

Prescribed Chemical Potential

The **Prescribed Chemical Potential** node adds a Dirichlet boundary condition where the chemical potential is prescribed. Only boundaries adjacent to domains where a Chemical Potential node has been assigned are allowed.

PRESCRIBED CHEMICAL POTENTIAL

Individually specify the value of the chemical potential for each species. Select the checkbox for the **Species** to specify its value, and then enter a value or expression in the corresponding field.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **Prescribed Chemical Potential**

Ribbon

Physics tab with **Transport in Solids** selected:

Boundaries > **Prescribed Chemical Potential**

Prescribed Transported Quantity

The **Prescribed Transported Quantity** node adds a Dirichlet boundary condition for the species.

PRESCRIBED TRANSPORTED QUANTITY

Specify the value of the dependent variable for each species. Select the checkbox for the **Species** to specify its value, and then enter a value or expression in the corresponding field. The SI unit depends on the selection done in Transported Quantity.

ADVANCED

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the prescribed value in the reference (material frame) or in the deformed (spatial frame) configuration. They are related by the volume ratio.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **Prescribed Transported Quantity**

Ribbon

Physics tab with **Transport in Solids** selected:

Boundaries > **Prescribed Transported Quantity**

Periodic Condition

The **Periodic Condition** node can be used to define periodicity for the transport between two sets of boundaries. The node prescribes continuity or a specified jump for the transported species and conservation of the flux between the "source" and the "destination" side, respectively. The direction in which the species is transported is dictated by the transport equations in the adjacent domains, and not by the "source" and "destination" names.

BOUNDARY SELECTION

If the node is activated on more than two boundaries, the feature tries to detect two distinct surfaces, each made up of one or more connected boundaries.

DESTINATION SELECTION

For more complex geometries, it might be necessary to specify the destination selection manually. To do so, right-click the **Periodic Condition** node and choose **Manual Destination Selection**. You can then specify the boundaries that constitute the destination surfaces in the **Destination Selection** section.

TRANSPORTED QUANTITY

Click the checkbox to activate the **Periodic Condition** for each transported species. Specify the value for the jump of the species concentration between the "source" and "destination" sides. By default there is no jump in concentration, $\Delta c_i = 0$.

ADVANCED

This setting is only available for deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the prescribed value in the reference (material frame) or in the deformed (spatial frame) configuration. They are related by the volume ratio.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. This section contains settings for specifying the type of constraint and whether to use a pointwise constraint, a weak constraint, or a constraint enforced by Nitsche's method. See Constraint Settings in the *COMSOL Multiphysics Reference Manual* for more information.

ORIENTATION OF SOURCE

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For information about the **Orientation of Source** section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

ORIENTATION OF DESTINATION

This section appears if the setting for **Transform to intermediate map** in the **Orientation of Source** section is changed from the default value, **Automatic**, and **Advanced Physics Options** is selected in the **Show More Options** dialog. For information about the **Orientation of Destination** section, see Orientation of Source and Destination in the *COMSOL Multiphysics Reference Manual*.

MAPPING BETWEEN SOURCE AND DESTINATION

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. For information about the **Mapping Between Source and Destination** section, see Mapping Between Source and Destination in the *COMSOL Multiphysics Reference Manual*.

See the Periodic Condition section in the Equation-Based Modeling chapter in the *COMSOL Multiphysics Reference Manual*.

Solid

The **Solid** node adds the equations to model the Fickian diffusion in solids. The node allows for specifying the diffusion properties of each species.

$$\Gamma_i = -D_i \nabla c_i$$

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains all applicable coordinate systems in the component. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data presented in a local system. The coordinate system must have orthonormal coordinate axes, and be defined in the material frame. Many of the possible subnodes inherit the coordinate system settings.

DIFFUSION

Use this section to specify the diffusion coefficient for each species.

From the **Material** list select **User defined** or a material available in the model to select a diffusion coefficient.

Enter the **Diffusion coefficient**, D_c, for each of the species.

For **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components. When anisotropic inputs are used, the axis orientations are given by the Coordinate System Selection.

MOBILITY

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog. The mobility is used by the Shrinkage and Swelling multiphysics coupling to compute stress-driven fluxes.

Select the **Mobility model** — **Linear**, **Quadratic**, or **User defined**.

When **Linear** is selected as the **Mobility model**, the mobility of each of the species is linearly related to the concentration

$$M_c = \frac{D_c}{RT}c$$

When **Quadratic** is selected as the **Mobility model**, enter the **Maximum concentration** c_{max} for each of the species. The default value is `Inf` mol/m^3. The mobility of each of the species is defined as

$$M_c = \frac{D_c}{RT}\left(1 - \frac{c}{c_{max}}\right)c$$

For **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components of the **Mobility coefficient**. When anisotropic inputs are used, the axis orientations are given by the Coordinate System Selection.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the

diffusion coefficient values in the reference (material frame) or in the deformed (spatial frame) configuration. They are related as follows,

$$D_i = JF^{-1}d_iF^{-T}$$

LOCATION IN USER INTERFACE

Context Menus

Transport in Solids > **Solid**

Ribbon

Physics tab with **Transport in Solids** selected:

Domains > **Solid**

Flux

Use the **Flux** node to specify the mass flux into or out of the model domain through some of its boundaries. With this boundary condition, positive values correspond to flow into the model domain:

$$-\mathbf{N} \cdot \Gamma_i = \gamma_0$$

where $\mathbf{N}$ denotes the outward pointing normal of the boundary in the reference configuration.

INWARD FLUX

Enter a value or expression in the corresponding field for the normal flux γ_0. A positive value of γ_0 represents an inward mass flux, whereas a negative value represents an outward mass flux. The units are based on the geometric entity and the units of the dependent variables specified in Transported Quantity.

ADVANCED

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the flux in the reference (material frame) or in the deformed (spatial frame) configuration.

LOCATION IN USER INTERFACE

Context Menus
Transport in Solids > **Flux**

Ribbon
Physics tab with **Transport in Solids** selected:

Boundaries > **Flux**

Source

The **Source** node describes generation within the domain of the dependent variable for the selected species. Specify the source contribution for each species in the corresponding field. You express addition or removal with a positive or negative value, respectively.

SOURCE

Enter a value or expression in the corresponding field for the source S_i for species i. The SI unit depends on the selection done in Transported Quantity.

ADVANCED

This setting is only available when the model includes deforming or rotating domains, for example when using a **Moving Mesh** node or a Structural Mechanics interface. From the **Frame** list, select **Reference configuration** or **Deformed configuration** to specify the source in the material frame or in the spatial frame. They are related by the volume ratio.

LOCATION IN USER INTERFACE

Context Menus
Transport in Solids > **Source**

Ribbon
Physics tab with **Transport in Solids** selected:

Domains > **Source**

Symmetry

The **Symmetry** node can be used to represent boundaries where the species distribution is symmetric, that is, where there is no flux across the boundary. This boundary

condition is identical to that of the No Flux node. It can be applied on exterior boundaries only.

LOCATION IN USER INTERFACE

Context Menus
Transport in Solids > **Symmetry**

Ribbon
Physics tab with **Transport in Solids** selected:

Boundaries > **Symmetry**

Electrode Surface Coupling

Use the **Electrode Surface Coupling** node to specify the mass flux into or out of the model domain based on current densities specified in an Electrochemistry interface.

The flux is proportional to the current densities and the stoichiometric coefficients according to Faraday's law. For details, see the Electrode Surface Coupling section in the *Structural Mechanics Theory* chapter.

MODEL INPUTS

Enter a value or expression for the **Coupled reaction** i_{loc}.

STOICHIOMETRIC COEFFICIENTS

Enter a value or expression in the corresponding field for the **Number of participating electrons** n and for the **Stoichiometric coefficient** ν_i for species i. The following convention is used: The stoichiometric coefficient of the reduced species ν_{red} is set to a positive value and the stoichiometric coefficient of the oxidized species ν_{ox} is set to a negative value.

LOCATION IN USER INTERFACE

Context Menus
Transport in Solids > **Electrode Surface Coupling**

Ribbon
Physics tab with **Transport in Solids** selected:

Boundaries > **Electrode Surface Coupling**

	The **Electrode Surface Coupling** node is available in a limited set of add-on products. See www.comsol.com/products/specifications/ for details.

17

Multiphysics Interfaces and Couplings

The Structural Mechanics Module contains predefined multiphysics interfaces to facilitate easy setup of models with the most commonly occurring couplings. When adding a multiphysics interface, two or more physics interfaces are added to the model together with one or more multiphysics couplings.

Given that your model already contains two interfaces that support a certain multiphysics coupling, you can also add this coupling separately. Some multiphysics couplings can only be added this way and are not part of a multiphysics interface.

In this chapter:

- The Thermal Stress, Solid Interface
- The Thermal Stress, Shell Interface
- The Thermal Stress, Layered Shell Interface
- The Thermal Stress, Membrane Interface
- The Joule Heating and Thermal Expansion Interface
- The Phase-Field Damage Interface
- The Phase-Field Damage, Explicit Dynamics Interface

- Overview of Fluid–Structure Interaction Interfaces
  - The Fluid–Solid Interaction Interface
  - The Fluid–Solid Interaction, Fixed Geometry Interface
  - The Fluid–Shell Interaction Interface
  - The Fluid–Shell Interaction, Fixed Geometry Interface
  - The Fluid–Membrane Interaction Interface
  - The Fluid–Membrane Interaction, Fixed Geometry Interface
  - The Fluid–Pipe Interaction, Fixed Geometry Interface
  - The Solid–Thin-Film Damping Interface
  - The Shell–Thin-Film Damping Interface
  - The Fluid–Solid Interaction, Viscoelastic Flow Interface
  - The Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry Interface
  - The Fluid–Solid Interaction, Conjugate Heat Transfer Interface
  - The Fluid–Solid Interaction, Two-Phase Flow, Phase Field Interface
  - The Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry Interface
- Overview of Poroelasticity Interfaces
  - The Poroelasticity, Solid Interface
  - The Poroelasticity, Layered Shell Interface
  - The Unsaturated Poroelasticity Interface

- Overview of Electromagnetics–Structure Interaction Interfaces
  - The Electromechanics, Solid Interface
  - The Electromechanics, Shell Interface
  - The Electromechanics, Membrane Interface
  - The Electromechanics, Boundary Elements Interface
  - The Piezoelectricity, Solid Interface
  - The Piezoelectricity, Layered Shell Interface
  - The Piezoelectricity and Pyroelectricity Interface
  - The Electrostriction Interface
  - The Ferroelectroelasticity Interface
  - The Magnetomechanics, Solid Interface
  - The Magnetomechanics, Shell Interface
  - The Magnetomechanics, Membrane Interface
  - The Magnetomechanics, No Currents, Solid Interface
  - The Magnetomechanics, No Currents, Shell Interface
  - The Magnetomechanics, No Currents, Membrane Interface
  - The Magnetic–Elastic Interaction in Rotating Machinery Interface
  - The Piezomagnetism Interface
  - The Nonlinear Magnetostriction Interface

- Multiphysics Couplings
  - Thermal Expansion
  - Thermal Expansion, Thin Layer
  - Thermal Expansion, Layered
  - Hygroscopic Swelling
  - Shrinkage and Swelling
  - Piezoelectricity
  - Piezoelectricity, Layered
  - Electrostriction
  - Electromechanics, Solid
  - Electromechanics, Boundary
  - Electromechanics, Boundary Elements
  - Magnetomechanics, Solid
  - Magnetomechanics, Boundary
  - Magnetic Forces, Rotating Machinery
  - Piezomagnetism
  - Nonlinear Magnetostriction
  - Magnetostriction
  - Fluid–Structure Interaction
  - Fluid–Structure Interaction, Pair
  - Fluid–Pipe Interaction
  - Structure–Thin-Film Flow Interaction
  - Shell–Thin-Film Flow Interaction
  - Phase-Field Damage (Multiphysics Coupling)

- Couplings Between Structural Mechanics Interfaces
  - Solid–Thin Structure Connection
  - Solid–Beam Connection
  - Shell–Beam Connection
  - Layered Shell–Structure Transition
  - Layered Shell–Structure Cladding
  - Layered Shell–Shell Connection
  - Structure–Pipe Connection
  - Embedded Reinforcement
  - Beam Cross Section–Beam Coupling
  - Beam–Beam Cross Section Coupling

The Thermal Stress, Solid Interface

The **Thermal Stress, Solid** ( ) interface combines a Solid Mechanics interface with a Heat Transfer in Solids interface. The coupling occurs on the domain level, where the temperature from the heat transfer interface acts as a thermal load for the solid mechanics interface, causing thermal expansion. Optionally, effects from thermoelastic damping and inelastic dissipation can be used as heat sources in the heat transfer analysis.

When a predefined **Thermal Stress, Solid** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Solid Mechanics** and **Heat Transfer in Solids** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, containing the multiphysics coupling Thermal Expansion.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Heat Transfer in Solids interface is described in The Heat Transfer in Solids Interface.

For information about the constitutive equations including thermal expansion in the section dealing with the theory background, see Structural Mechanics Theory.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings (for example, **Thermal Stress, Solid**), specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Solid Mechanics and Heat Transfer in Solids interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose **Thermal**

Expansion from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-1: MODIFIED SETTINGS FOR A THERMAL STRESS, SOLID INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior the **Structural transient behavior** is set to **Quasi static**.
Thermal Expansion	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Heat Transfer in Solids interfaces are preselected in the Coupled Interfaces section.

- *Fuel Cell Bipolar Plate*: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/bipolar_plate**
- *Thermal Expansion in a MEMS Device*: Application Library path **MEMS_Module/Sensors/thermal_expansion**

The Thermal Stress, Shell Interface

The **Thermal Stress, Shell** ( ) interface combines a Shell interface with a Heat Transfer in Shells interface. The coupling occurs on boundaries, where the temperature from the heat transfer interface acts as a thermal load for the shell interface, causing thermal expansion. Optionally, effects from thermoelastic damping and inelastic dissipation can be used as heat sources in the heat transfer analysis.

When a predefined **Thermal Stress, Shell** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Shell** and **Heat Transfer in Shells** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, containing the multiphysics coupling Thermal Expansion, Layered.

On the Constituent Physics Interfaces

- The Shell interface is described in The Shell and Plate Interfaces.
- The Heat Transfer in Shells interface is described in The Heat Transfer in Shells Interface.

For information about the constitutive equations including thermal expansion in the section dealing with the theory background, see Theory for the Shell and Plate Interfaces.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Thermal Stress, Shell**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Shell and Heat Transfer in Shells interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-2: MODIFIED SETTINGS FOR A THERMAL STRESS, SHELL INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Shell	For the Shell interface, under Structural Transient Behavior the **Structural transient behavior** is set to **Quasi static**.
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to **Linear**.
Thermal Expansion, Layered	The **Boundary Selection** is the same as that of the participating physics interfaces. The corresponding Shell and Heat Transfer in Shells interfaces are preselected in the Coupled Interfaces section.

The Thermal Stress, Layered Shell Interface

The **Thermal Stress, Layered Shell** ( ) interface combines a Layered Shell interface with a Heat Transfer in Shells interface. The coupling occurs on boundaries, where the temperature from the heat transfer interface acts as a thermal load for the Layered Shell interface, causing thermal expansion. Optionally, effects from thermoelastic damping and inelastic dissipation can be used as heat sources in the heat transfer analysis.

When a predefined **Thermal Stress, Layered Shell** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Layered Shell** and **Heat Transfer in Shells** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added which automatically includes the multiphysics coupling node Thermal Expansion, Layered

On the Constituent Physics Interfaces

- The Layered Shell interface is described in The Layered Shell Interface.
- The Heat Transfer in Shells interface is described in The Heat Transfer in Shells Interface.

For information about the constitutive equations including thermal expansion in the section dealing with the theory background, see Theory for the Layered Shell Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Thermal Stress, Layered Shell**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Layered Shell and Heat Transfer in Shells interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the

available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-3: MODIFIED SETTINGS FOR A THERMAL STRESS, LAYERED SHELL INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Layered Shell	For the Layered Shell interface, under Structural Transient Behavior the **Structural transient behavior** is set to **Quasi static**.
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to **Linear**.
Thermal Expansion, Layered	The **Boundary Selection** is the same as that of the participating physics interfaces. The corresponding Layered Shell and Heat Transfer in Shells interfaces are preselected in the Coupled Interfaces section.

The Thermal Stress, Membrane Interface

The **Thermal Stress, Membrane** ( ) interface combines a Membrane interface with a Heat Transfer in Shells interface. The coupling occurs on boundaries, where the temperature from the heat transfer interface acts as a thermal load for the Membrane interface, causing thermal expansion. Optionally, effects from thermoelastic damping and inelastic dissipation can be used as heat sources in the heat transfer analysis.

When a predefined **Thermal Stress, Membrane** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Membrane** and **Heat Transfer in Shells** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added which automatically includes the multiphysics coupling node Thermal Expansion, Layered.

On the Constituent Physics Interfaces

- The Membrane interface is described in The Membrane Interface.
- The Heat Transfer in Shells interface is described in The Heat Transfer in Shells Interface.

For information about the constitutive equations including thermal expansion in the section dealing with the theory background, see Theory for the Membrane Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings (for example, **Thermal Stress, Membrane**), specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Membrane and Heat Transfer in Shells interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-4: MODIFIED SETTINGS FOR A THERMAL STRESS, MEMBRANE INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Membrane	For the Membrane interface, under Structural Transient Behavior the **Structural transient behavior** is set to **Quasi static**.
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to **Linear**.
Thermal Expansion, Layered	The **Boundary Selection** is the same as that of the participating physics interfaces. The corresponding Membrane and Heat Transfer in Shells interfaces are preselected in the Coupled Interfaces section.

The Joule Heating and Thermal Expansion Interface

The **Joule Heating and Thermal Expansion** interface ( ) combines thermal, electric, and structural multiphysics effects. The predefined interaction adds the electromagnetic losses from the electric current as a heat source. In addition, the temperature from the Heat Transfer in Solids interface acts as a thermal load for the Solid Mechanics interface, causing thermal expansion. Optionally, effects from thermoelastic damping and inelastic dissipation can be used as heat sources in the heat-transfer analysis.

You can use this multiphysics coupling for coupled thermal, electrical, and structural analysis of, for example, the movement of some actuator, where an electric current causes a temperature increase, which in turn leads to a displacement through thermal expansion.

When a predefined **Joule Heating and Thermal Expansion** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, Solid Mechanics, Electric Currents, and Heat Transfer in Solids interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, containing the multiphysics coupling nodes Thermal Expansion and Electromagnetic Heating.

On the Constituent Physics Interfaces

- The Electric Currents interface is described in The Electric Currents Interface.
- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Heat Transfer in Solids interface is described in The Heat Transfer in Solids Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Joule Heating and Thermal Expansion**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Solid Mechanics, Electric Currents, and Heat Transfer in Solids interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can then choose from the available coupling features, **Thermal Expansion**, and **Electromagnetic Heating**, but the modified settings are not included.

TABLE 17-5: MODIFIED SETTINGS FOR A JOULE HEATING AND THERMAL EXPANSION INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior, the **Structural transient behavior** is set to **Quasi static**.
Electric Currents	No changes.
Heat Transfer in Solids	No changes.
Thermal Expansion	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Heat Transfer in Solids interfaces are preselected in the Thermal Expansion section.
Electromagnetic Heating	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Electric Currents and Heat Transfer in Solids interfaces are preselected in the Electromagnetic Heating section.

If you have an add-on module, such as the Heat Transfer Module or AC/DC Module, there are additional specialized physics nodes available and described in the individual module documentation.

Overview of Fluid–Structure Interaction Interfaces

The **Model Wizard** and the **Add Physics** window contain a number of multiphysics interfaces for modeling fluid–structure interaction (FSI), all available under the **Fluid–Structure Interaction** branch ( ). The multiphysics interfaces for FSI are described in the subsequent sections:

- The Fluid–Solid Interaction Interface: For fluid–structure interaction between a fluid and a solid.
- The Fluid–Solid Interaction, Fixed Geometry Interface: For fluid–structure interaction between a fluid and a solid where the displacements of the solid are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.
- The Fluid–Shell Interaction Interface: For fluid–structure interaction between a fluid and a shell structure.
- The Fluid–Shell Interaction, Fixed Geometry Interface: For fluid–structure interaction between a fluid and a shell structure where the displacements of the shell are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.
- The Fluid–Membrane Interaction Interface: For fluid–structure interaction between a fluid and a membrane structure.
- The Fluid–Membrane Interaction, Fixed Geometry Interface: For fluid–structure interaction between a fluid and a membrane structure where the displacements of the membrane are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.
- The Fluid–Pipe Interaction, Fixed Geometry Interface: This multiphysics interface provides a connection between the Pipe Flow interface and the Pipe Mechanics interface. The coupling is unidirectional, so that various types of loads from the fluid loads can be transferred to structural analysis.
- The Solid–Thin-Film Damping Interface: For fluid–structure interaction between a thin-film fluid and a solid under isothermal conditions.

- The Shell–Thin-Film Damping Interface: For fluid–structure interaction between a thin-film fluid and a shell under isothermal conditions.
- The Fluid–Solid Interaction, Viscoelastic Flow Interface: For fluid–structure interaction between a solid and viscoelastic fluids.
- The Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry Interface: For fluid–structure interaction between a solid and viscoelastic fluids where the displacements of the solid are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.
- The Fluid–Multibody Interaction Interface: For fluid–structure interaction between a fluid and mechanical parts. See Multiphysics Couplings in the *Multibody Dynamics Module User's Guide*.
- The Fluid–Multibody Interaction, Assembly Interface: For fluid–structure interaction between a fluid and mechanical parts. An ALE formulation is used for incorporating the geometrical changes of the fluid domain. See Multiphysics Couplings in the *Multibody Dynamics Module User's Guide*.
- The Fluid–Solid Interaction, Conjugate Heat Transfer Interface: For fluid–structure interaction between a fluid and a solid under nonisothermal conditions.
- The Fluid–Solid Interaction, Two-Phase Flow, Phase Field Interface: For fluid–structure interaction between two immiscible fluids and a deformable solid. An ALE formulation is used for incorporating the geometrical changes of the fluid domain. The interface between the two fluids is tracked by the phase field method.
- The Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry Interface: For fluid–structure interaction between two immiscible fluids and a solid where the displacements of the solid are assumed to be small enough to be considered as fixed during the interaction. The interface between the two fluids is tracked by the phase field method.

The Fluid–Solid Interaction Interface

Using the **Fluid–Solid Interaction** ( ) interface, you can model phenomena where a fluid and a deformable solid affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain.

Using a stationary or a time-dependent study, the **Fluid–Solid Interaction** interface models bidirectional coupling between solids and fluids.

You add a predefined **Fluid–Solid Interaction** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Solid Mechanics** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, containing the multiphysics coupling Fluid–Structure Interaction.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The **Deforming Domain** feature is described in Moving Mesh Features.

In 2D, plane stress or plane strain assumptions can be selected in the Solid Mechanics interface. Only the plane strain (default) should be used together with the FSI coupling.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if a Moving Mesh, a Laminar Flow, and a Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

TABLE 17-6: MODIFIED SETTINGS FOR THE FLUID–SOLID INTERACTION INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Moving Mesh	A Deforming Domain is added, where the **Selection** is set to **Manual** in the Domain Selection section.
Solid Mechanics	For the Solid Mechanics interface, under Discretization, the **Displacement field** is set to **Quadratic Lagrange**.
Laminar Flow	No changes.
Fluid–Structure Interaction	The **Selection** is set to **All boundaries** in the Boundary Selection section. The corresponding Solid Mechanics and Laminar Flow interfaces are preselected in the Coupled Interfaces section. The **Fixed geometry coupling type** is set to **Fully coupled** in the Fixed Geometry section.

The Fluid–Solid Interaction, Fixed Geometry Interface

The **Fluid–Solid Interaction, Fixed Geometry** ( ) interface can be used to model phenomena where a fluid and a deformable solid structure affect each other. Both the fluid loading on the structure and the structural velocity transmission to the fluid can be taken into account. The interface models situations where the displacements of the solid are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.

You add a predefined **Fluid–Solid Interaction, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Solid Mechanics** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, containing the multiphysics coupling Fluid–Structure Interaction.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Solid Mechanics interface is described in The Solid Mechanics Interface.

	You can add a **Stationary, One-Way FSI** or **Time Dependent, One-Way FSI** studies. See Stationary and Time Dependent One-Way FSI Studies for Fluid–Multibody Interaction in the *COMSOL Multiphysics Reference Manual* for more information.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

TABLE 17-7: MODIFIED SETTINGS FOR THE FLUID–SOLID INTERACTION, FIXED GEOMETRY INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Discretization, the **Displacement field** is set to **Quadratic Lagrange**.
Laminar Flow	No changes.
Fluid–Structure Interaction	The **Selection** is set to **All boundaries** in the Boundary Selection section. The corresponding Solid Mechanics and Laminar Flow interfaces are preselected in the Coupled Interfaces section. The **Fixed geometry coupling type** is set to **Fully coupled** in the Fixed Geometry section.

COUPLING FEATURES

The Fluid–Structure Interaction coupling feature is used to couple the interfaces.

The Fluid–Shell Interaction Interface

Using the **Fluid–Shell Interaction** ( ) interface, you can model phenomena where a fluid and a deformable shell affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain.

Using a stationary or a time-dependent study, the **Fluid–Shell Interaction** interface models bidirectional coupling between solids and fluids.

You add a predefined **Fluid–Shell Interaction** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Shell** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Shell interface is described in The Shell and Plate Interfaces.
- The **Deforming Domain** feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Shell interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Features

The Fluid–Structure Interaction coupling feature is used to couple the interfaces.

Physics Interface Features

- The available physics features for The Single-Phase Flow, Laminar Flow Interface are listed in the section Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow in the *COMSOL Multiphysics Reference Manual*.
- The available physics features for The Shell and Plate Interfaces are listed in the Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces section.

The Fluid–Shell Interaction, Fixed Geometry Interface

The **Fluid–Shell Interaction, Fixed Geometry** ( ) interface can be used to model phenomena where a fluid and a deformable shell structure affect each other. Both the fluid loading on the structure and the structural velocity transmission to the fluid can be taken into account. The interface models situations where the displacements of the shell are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.

You add a predefined **Fluid–Shell Interaction, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Shell** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Shell interface is described in The Shell and Plate Interfaces.

	You can add a **Stationary, One-Way FSI** or **Time Dependent, One-Way FSI** studies. See Stationary and Time Dependent One-Way FSI Studies for Fluid–Multibody Interaction in the *COMSOL Multiphysics Reference Manual* for more information.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Shell interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Fluid–Structure Interaction coupling feature is used to couple the interfaces.

The Fluid–Membrane Interaction Interface

Using the **Fluid–Membrane Interaction** ( ) interface, you can model phenomena where a fluid and a deformable membrane affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain.

Using a stationary or a time-dependent study, the **Fluid–Membrane Interaction** interface models bidirectional coupling between solids and fluids.

You add a predefined **Fluid–Membrane Interaction** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Membrane** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Membrane interface is described in The Membrane Interface.
- The **Deforming Domain** feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Membrane interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Fluid–Structure Interaction coupling feature is used to couple the interfaces.

The Fluid–Membrane Interaction, Fixed Geometry Interface

The **Fluid–Membrane Interaction, Fixed Geometry** ( ) interface can be used to model phenomena where a fluid and a deformable membrane structure affect each other. Both the fluid loading on the structure and the structural velocity transmission to the fluid can be taken into account. The interface models situations where the displacements of the membrane are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.

You add a predefined **Fluid–Membrane Interaction, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Membrane** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Membrane interface is described in The Membrane Interface.

	You can add a **Stationary, One-Way FSI** or **Time Dependent, One-Way FSI** studies. See Stationary and Time Dependent One-Way FSI Studies for Fluid–Multibody Interaction in the *COMSOL Multiphysics Reference Manual* for more information.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Membrane interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Features

The Fluid–Structure Interaction coupling feature is used to couple the interfaces.

Physics Interface Features

- The available physics features for The Single-Phase Flow, Laminar Flow Interface are listed in the section Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow in the *COMSOL Multiphysics Reference Manual*.
- The available physics features for The Membrane Interface are listed in the Boundary, Edge, Point, and Pair Nodes for the Membrane Interface section.

The Fluid–Pipe Interaction, Fixed Geometry Interface

The **Fluid–Pipe Interaction, Fixed Geometry** ( ) interface can be used to model problems where the forces from the fluid in a piping system act as structural loads. The structural deformations do not affect the flow.

You add a predefined **Fluid–Pipe Interaction, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Pipe Flow** and **Pipe Mechanics** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling node Fluid–Pipe Interaction.

On the Constituent Physics Interfaces

- The Pipe Flow interface is described in The Pipe Flow Interface in the *Pipe Flow Module User's Guide*.
- The Pipe Mechanics interface is described in The Pipe Mechanics Interface.

The Solid–Thin-Film Damping Interface

The **Solid–Thin-Film Damping** ( ) interface can be used to model phenomena where a thin-film fluid and a deformable solid affect each other. The fluid can be either a liquid or a gas, with the possibility to include cavitation in liquids.

Use the **Solid–Thin-Film Damping** interface to apply boundary loads to the surface of a resonator that result from squeeze-film or slide-film damping. Examples of common situations in which these types of damping are appropriate are parallel-plate capacitive and comb-drive actuators for squeeze-film and slide-film damping, respectively.

You add a predefined **Solid–Thin-Film Damping** interface from the **Fluid–Structure Interaction** ( ) group in the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Solid Mechanics** and **Thin-Film Flow** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Structure Thin-Film Flow Interaction**.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface
- The Thin-Film Flow interface is described in The Thin-Film Flow Interfaces.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if Thin-Film Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Structure–Thin-Film Flow Interaction coupling feature is used to couple the interfaces.

See Theory for the Thin-Film Flow Interfaces in the *CFD Module User's Guide*.

The Shell–Thin-Film Damping Interface

The **Shell–Thin-Film Damping** ( ) interface can be used to model phenomena where a thin-film fluid and a shell affect each other. The fluid can be either a liquid or a gas, with the possibility to include cavitation in liquids.

Using a frequency domain or a time-dependent study, the interface can model fluid-induced damping in shells.

You add a predefined **Shell–Thin-Film Damping** interface from the **Fluid–Structure Interaction** ( ) group in the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Shell** and **Thin-Film Flow** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Shell–Thin-Film Flow Interaction**.

On the Constituent Physics Interfaces

- The Shell interface is described in The Shell and Plate Interfaces.
- The Thin-Film Flow interface is described in The Thin-Film Flow Interfaces.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if Thin-Film Flow and Shell interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Shell–Thin-Film Flow Interaction coupling feature is used to couple the interfaces.

See Theory for the Thin-Film Flow Interfaces in the *CFD Module User's Guide*.

The Fluid–Solid Interaction, Viscoelastic Flow Interface

Using the **Fluid–Solid Interaction, Viscoelastic Flow** ( ) interface, you can model phenomena where a deformable solid and a fluid exhibiting a combination of viscous and elastic behavior affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain.

Using a time-dependent study, the **Fluid–Solid Interaction, Viscoelastic Flow** interface models bidirectional coupling between solids and fluids.

You add a predefined **Fluid–Solid Interaction, Viscoelastic Flow** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Viscoelastic Flow** and **Solid Mechanics** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added to the Model Builder, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

The **Viscoelastic Flow** interface solves the continuity equation for conservation of mass and the Navier–Stokes equations for the conservation of momentum with an additional elastic stress term. To calculate the extra stress tensor, a number of constitutive relations are available.

The solid domain has all functionality available in the Solid Mechanics interface, including contact conditions and also nonlinear materials if the Nonlinear Structural Materials Module or Geomechanics Module is available.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Viscoelastic Flow interface is described in The Viscoelastic Flow Interface.
- The **Deforming Domain** feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Fluid–Structure Interaction feature is used to couple the interfaces.

See Theory for the Thin-Film Flow Interfaces in the *CFD Module User's Guide*.

The Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry Interface

Using the **Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry** ( ) interface, you can model phenomena where a deformable solid and a fluid exhibiting a combination of viscous and elastic behavior affect each other. Both the fluid loading on the structure and the structural velocity transmission to the fluid can be taken into account. The interface models situations where the displacements of the solid are assumed to be small enough for the geometry of the fluid domain to be considered as fixed during the interaction.

You add a predefined **Fluid–Solid Interaction, Viscoelastic Flow, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Viscoelastic Flow** and **Solid Mechanics** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added to the Model Builder, which automatically includes the multiphysics coupling feature **Fluid–Structure Interaction**.

The **Viscoelastic Flow** interface solves the continuity equation for conservation of mass and the Navier–Stokes equations for the conservation of momentum with an additional elastic stress term. To calculate the extra stress tensor, a number of constitutive relations are available.

The solid domain has all functionality available in the Solid Mechanics interface, including contact conditions and also nonlinear materials if the Nonlinear Structural Materials Module or Geomechanics Module is available.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Viscoelastic Flow interface is described in The Viscoelastic Flow Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Fluid–Structure Interaction feature is used to couple the interfaces.

The Fluid–Solid Interaction, Conjugate Heat Transfer Interface

Using the **Fluid–Solid Interaction, Conjugate Heat Transfer** ( ) interface, you can model phenomena where a fluid and a deformable solid affect each other at the same time as heat is transferred through and between the domains. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain.

Using a stationary or a time-dependent study, the **Fluid–Solid Interaction, Conjugate Heat Transfer** interface models bidirectional coupling between solids and fluids. You can, however, by modifying the solver sequences also model various types of unidirectional couplings.

You add this interface from the **Conjugate Heat Transfer** ( ) subgroup within the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. Three physics interfaces are then added to the Model Builder: Laminar Flow, Heat Transfer in Solids and Fluids, and Solid Mechanics.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling nodes **Fluid–Structure Interaction**, **Nonisothermal Flow**, and **Thermal Expansion**.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

On the Constituent Physics Interfaces

- The Laminar Flow interface is described in The Single-Phase Flow, Laminar Flow Interface. When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.
- The Solid Mechanics interface is described in The Solid Mechanics Interface.

	In 2D, plane stress or plane strain assumptions can be selected in the Solid Mechanics interface. Only the plane strain (default) should be used together with an FSI coupling.

- The Heat Transfer in Solids and Fluids interface is described in The Heat Transfer in Solids and Fluids Interface. A Solid feature is active by default in all domains, and a Fluid feature is added without a domain selection. All functionality for including other domain types is also available.
- The Deforming Domain feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Fluid–Structure Interaction, Nonisothermal Flow, and Thermal Expansion coupling features are used to couple the interfaces.

The Fluid–Solid Interaction, Two-Phase Flow, Phase Field Interface

Using the **Fluid–Solid Interaction, Two-Phase Flow, Phase Field** ( ) interface, you can model phenomena where two immiscible fluids and a deformable solid affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. Arbitrary Lagrangian–Eulerian Formulation (ALE) is used for incorporating the geometrical changes of the fluid domain. The interface between the two fluids is tracked by solving two additional transport equations, one for the phase field variable and one for the mixing energy density.

Using a time-dependent study, the **Fluid–Solid Interaction, Two-Phase Flow, Phase Field** interface models bidirectional coupling between solids and the fluids. Simulations using the interface are always time-dependent since the position of the interface between the two fluids is almost always dependent of its history.

You add a predefined **Fluid–Solid Interaction, Two-Phase Flow, Phase Field** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow**, **Solid Mechanics**, and **Phase Field** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added to the Model Builder, which automatically includes the multiphysics coupling features **Fluid–Structure Interaction** and **Two-Phase Flow, Phase Field**.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see Deforming Fluid Domains.

The flow regime can be laminar or turbulent (if the CFD Module is available). The solid domain has all functionality available in the Solid Mechanics interface, including contact conditions and also nonlinear materials if the Nonlinear Structural Materials Module or Geomechanics Module is available.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.

- The Two-Phase Flow interface is described in The Two-Phase Flow, Level Set and Phase Field Interfaces.
- The **Deforming Domain** feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Two-Phase Flow, Phase Field Coupling Feature and the Fluid–Structure Interaction feature are used to couple the interfaces.

The Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry Interface

Using the **Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry** ( ) interface, you can model phenomena where two immiscible fluids and a deformable solid affect each other. The interface includes a predefined condition for the interaction at the fluid–solid boundaries. The interface between the two fluids is tracked by solving two additional transport equations, one for the phase field variable and one for the mixing energy density.

Using a time-dependent study, the **Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry** interface models bidirectional coupling between solids and the fluids. Simulations using the interface are always time-dependent since the position of the interface between the two fluids is almost always dependent of its history.

You add a predefined **Fluid–Solid Interaction, Two-Phase Flow, Phase Field, Fixed Geometry** interface from the **Fluid–Structure Interaction** ( ) group in either the **Structural Mechanics** branch ( ) or the **Fluid Flow** branch ( ) of the **Model Wizard** or **Add Physics** windows. **Laminar Flow**, **Solid Mechanics**, and **Phase Field** interfaces are then added to the Model Builder.

In addition, the **Multiphysics** node is added to the Model Builder, which automatically includes the multiphysics coupling features **Fluid–Structure Interaction** and **Two-Phase Flow, Phase Field**.

The flow regime can be laminar or turbulent (if the CFD Module is available). The solid domain has all functionality available in the Solid Mechanics interface, including contact conditions and also nonlinear materials if the Nonlinear Structural Materials Module or Geomechanics Module is available.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.

- The Two-Phase Flow interface is described in The Two-Phase Flow, Level Set and Phase Field Interfaces.

You can add a **Stationary, One-Way FSI** or **Time Dependent, One-Way FSI** studies. See Stationary and Time Dependent One-Way FSI Studies for Fluid–Multibody Interaction in the *COMSOL Multiphysics Reference Manual* for more information.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Laminar Flow and Solid Mechanics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

COUPLING FEATURES

The Two-Phase Flow, Phase Field Coupling Feature and the Fluid–Structure Interaction feature are used to couple the interfaces.

Overview of Poroelasticity Interfaces

The **Model Wizard** and the **Add Physics** window contain a number of multiphysics interfaces for modeling poroelasticity, all available under the **Poroelasticity** branch ( ). The multiphysics interfaces for poroelasticity are described in the subsequent sections:

- The Poroelasticity, Solid Interface: For poroelastic interaction between a fluid and a solid.
- The Poroelasticity, Layered Shell Interface: For poroelastic interaction between a fluid and a layered shell.
- The Unsaturated Poroelasticity Interface: For poroelastic interaction in partially saturated porous media.

The Poroelasticity, Solid Interface

The Poroelasticity, Solid Interface

The **Poroelasticity, Solid** interface ( ) combines a transient formulation of Darcy's law with a quasistatic formulation of Solid Mechanics under isothermal conditions. The coupling occurs on the domain level, where the pore pressure from the Darcy's Law interface acts as a load for the Solid Mechanics interface, causing swelling or shrinkage. Changes in volumetric strain affect the porosity, acting as a mass source or sink in Darcy's law.

When a predefined **Poroelasticity, Solid** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, the **Solid Mechanics** and **Darcy's Law** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the **Poroelasticity** multiphysics coupling.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Darcy's Law interface is described in The Darcy's Law Interface.

	To model orthotropic or anisotropic porous materials requires the Structural Mechanics Module. For information about the constitutive equations and theory background, see Structural Mechanics Theory.

The **Poroelasticity** multiphysics coupling handles both geometrically linear and nonlinear problems automatically according to the settings in material model used in the Solid Mechanics interface. In the default **Linear Elastic Material** node, geometric nonlinearity is controlled by the **Formulation** list. By default, it is set to **From study step**. Changing this setting to **Total Lagrangian** forces the checkbox **Include geometric nonlinearity** in the study step to be selected and grayed out.

Conversely, select **Geometrically linear** in the **Formulation** list to enforce a small strain formulation, even if the checkbox **Include geometric nonlinearity** is selected in the study. See Geometric Nonlinearity for the Solid Mechanics Interface in the *Structural Mechanics Module User's Guide*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Poroelasticity**, specific settings are included with the physics interfaces and the coupling features.

However, if the physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Solid Mechanics** and **Darcy's Law** interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-8: MODIFIED SETTINGS FOR A POROELASTICITY INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior, the **Structural transient behavior** is set to **Quasistatic**.
Darcy's Law	For the Darcy's Law interface, a **Poroelastic storage** model is added by default.
Poroelasticity	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Darcy's Law interfaces are preselected in the Coupled Interfaces section.

Failure of a Multilateral Well: Application Library path **Subsurface_Flow_Module/Poromechanics/multilateral_well**

Large Swelling in Polymer Hydrogels: Application Library path **Porous_Media_Flow_Module/Poromechanics/hydrogel_swelling**

Poroelasticity (Multiphysics Coupling)

The **Poroelasticity** multiphysics coupling node ( ) links the Solid Mechanics and Darcy's Law interfaces bidirectionally to account for poroelastic deformation.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `poro1`.

DOMAIN SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains to define the poroelasticity coupling, or select **All domains** as needed.

When **Poroelasticity** is added as an effect of adding a Poroelasticity interface, the selection is the same as for the participating physics interfaces.

Only domains that are active in the physics interfaces selected in the Coupled Interfaces section can be selected.

The **Poroelasticity** coupling handles geometric nonlinearity automatically in order to apply a strain formulation that is consistent with the geometric nonlinearity settings in the solid material model for each domain selected, see Geometric Nonlinearity for the Solid Mechanics Interface in the Structural Mechanics Modeling chapter for details.

COUPLED INTERFACES

This section defines the physics involved in the multiphysics coupling. The **Solid mechanics** and **Flow in porous media** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a multiphysics interface is selected in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the Poroelasticity node from a physics interface. If the physics interface is removed from the **Model Builder**, for example if **Darcy's Law** is deleted, then the **Flow in porous media** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the **Solid mechanics** or **Flow in porous media** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

POROELASTIC COUPLING PROPERTIES

In the **Poroelasticity model** list, choose between **Biot** and **Biphasic**. The Biot poroelasticity model (default) is commonly used in geotechnical applications where the compressibility of the solid and/or fluid phase is important. In contrast, the biphasic model assumes that both the solid and the fluid phases are incompressible, which is valid for soft materials like biological tissues and hydrogels. Note that even if both constituents are modeled as incompressible, the porous medium can still undergo large volume changes as a result of fluid inflow or outflow.

In the **Reference pressure level** list, choose **User defined** to enter a reference pressure p_{ref} (SI unit: Pa) at which the pore pressure does not affect the porous matrix deformation. The default value is 1 atm. If applicable, the list also contains options to inherit the value from another physics interface that announces the reference pressure level as a model input. For example, the reference pressure level can often be inherited from the

corresponding input in the Darcy's Law interface that is part of the Poroelasticity coupling.

For the **Biot–Willis coefficient** list, the default is to take its value **From material**. Choose **User defined** to enter a Biot–Willis coefficient α_B (dimensionless) in the text field. The user input is only required if the **Poroelasticity model** is set to **Biot**.

The Poroelasticity, Layered Shell Interface

The **Poroelasticity, Layered Shell** interface ( ) combines the **Layered Darcy's Law** formulation with the **Layered Shell** interface for structural mechanics. This multiphysics interface is only available for 3D simulations and is only active on all boundaries where a layered material is defined.

A layered material can be defined on the boundaries by adding one of the following nodes:

- **Single Layer Material**
- **Layered Material Link**, together with a **Layered Material** node
- **Layered Material Stack**, together with one or several **Layered Material** nodes
- **Material**, with the **Layer thickness** property specified

See Layered Material, Layered Material Link, Layered Material Stack, Layered Material Link (Subnode), and Single Layer Materials in the *COMSOL Multiphysics Reference Manual* for details on the definition of layered materials.

This multiphysics interface is only available when the Structural Mechanics Module, the Composite Materials Module and the Porous Media Flow Module are licensed. For a detailed overview of the functionality available in each product visit www.comsol.com/products/specifications/.

The coupling occurs on the boundary level, where the pore pressure from the Layered Darcy's Law interface acts as a load for the Layered Shell interface, causing swelling or shrinking. Changes in volumetric strain affect the pore space, acting as a mass source or sink in Darcy's Law.

When a predefined **Poroelasticity, Layered Shell** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Layered Shell** and **Layered Darcy's Law** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the **Poroelasticity, Layered** multiphysics coupling.

For the settings of each of the coupled interfaces see

- The Layered Darcy's Law Interface and
- The Layered Shell Interface in the *Structural Mechanics Module User's Guide*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, as in this case **Poroelasticity, Layered Shell**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Layered Shell** and **Layered Darcy's Law** interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-9: MODIFIED SETTINGS FOR A POROELASTICITY INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Layered Shell	For the Layered Shell interface, under Structural Transient Behavior the **Structural transient behavior** is set to **Quasistatic**.
Layered Darcy's Law	For the Layered Darcy's Law interface, under Discretization, the shape function order for the temperature is set to **Linear**. Also, a **Poroelastic Storage** model is added by default.
Poroelasticity, Layered	The corresponding Layered Shell and Layered Darcy's Law interfaces are preselected in the Coupled Interfaces section.

Poroelasticity, Layered (Multiphysics Coupling)

The **Poroelasticity, Layered** multiphysics coupling node ( ) links bidirectionally the Layered Shell and Layered Darcy's Law interfaces to account for poroelastic deformation on layered shells.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** is `poro`, followed by the number of defined multiphysics couplings, that is `poro1` for the first multiphysics coupling feature in the model.

BOUNDARY SELECTION

When **Poroelasticity, Layered** is added to the **Multiphysics** node as an effect of adding a Poroelasticity, Layered Shell interface, the selection is the same as for the participating physics interfaces. In the Boundary Selection menu you can choose between **Manual** and **All boundaries**, the latter being the default option. Note that only boundaries that are active in the physics interfaces selected in the **Coupled Interfaces** section can be selected.

COUPLED INTERFACES

This section defines the physics involved in the multiphysics coupling. The **Layered shell** and **Flow in porous media** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a multiphysics interface is selected in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the Poroelasticity node from a physics interface. If the physics interface is removed from the **Model Builder**, for

example **Darcy's Law** is deleted, then the **Flow in porous media** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the **Layered shell** or **Flow in porous media** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

SHELL PROPERTIES

Here you select on which layers of a layered material the physics interface should be active. By default, the **Use all layers** checkbox is selected. This means that all layers in all layered materials on the selected boundaries are used.

If you clear the **Use all layers** checkbox, you can select individual layers within a single layered material. This means that the physics interface is restricted to the boundaries on which a specific layered material is defined.

For a general description of layer selections, see The Shell Properties and Interface Selection Sections in the *Structural Mechanics Module User's Guide*.

POROELASTIC COUPLING PROPERTIES

In the **Reference pressure level** field, enter a reference pressure p_{ref} (SI unit: Pa) at which the pore pressure does not affect the porous matrix deformation. The default value is 1 atm.

For the **Biot–Willis coefficient** list, the default is to take its value **From material**. Choose **User defined** to enter a Biot–Willis coefficient α_B (dimensionless) in the text field.

The Unsaturated Poroelasticity Interface

The Unsaturated Poroelasticity Interface

The **Unsaturated Poroelasticity** interface ( ) combines a transient formulation of Moisture Transport in Solids with a quasistatic formulation of Solid Mechanics. The coupling occurs on the domain level, where the pore pressure from the Moisture Transport in Solids interface acts as a load for the Solid Mechanics interface, causing swelling or shrinking. Changes in volumetric strain affect the pore space, affecting the transport of moist air and liquid water.

When a predefined **Unsaturated Poroelasticity** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, the **Solid Mechanics** and **Moisture Transport in Solids** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the **Unsaturated Poroelasticity** multiphysics coupling.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Moisture Transport in Solids interface is described in The Moisture Transport in Solids Interface.

Theory for the Poroelasticity Interfaces.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Unsaturated Poroelasticity**, specific settings are included with the physics interfaces and the coupling features.

However, if the physics interfaces are added one at a time, followed by the coupling features, these predefined settings are not automatically included.

For example, if single **Solid Mechanics** and **Moisture Transport in Solids** interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the

available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-10: MODIFIED SETTINGS FOR A POROELASTICITY INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior, the **Structural transient behavior** is set to **Quasistatic**.
Moisture Transport in Solids	For the Moisture Transport in Solids interface, a **Porous Medium** node is added by default.
Unsaturated Poroelasticity	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Moisture Transport in Solids interfaces are preselected in the Coupled Interfaces section.

Moisture Transport in a Paperboard Roll: Application Library path **Structural_Mechanics_Module/Hygroscopic_Swelling/paperboard_roll**

Unsaturated Poroelasticity (Multiphysics Coupling)

The **Unsaturated Poroelasticity** multiphysics coupling node ( ) links the Moisture Transport in Solids interfaces bidirectionally with the Solid Mechanics interface to account for poroelastic deformation.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `unporo1`.

DOMAIN SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains to define the unsaturated poroelasticity coupling, or select **All domains** as needed.

When **Unsaturated Poroelasticity** is added, the selection is the same as for the participating physics interfaces.

Only domains that are active in the physics interfaces selected in the Coupled Interfaces section can be selected.

COUPLED INTERFACES

This section defines the physics involved in the multiphysics coupling. The **Solid mechanics** and **Moisture transport** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a multiphysics interface is selected in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the Unsaturated Poroelasticity node from a physics interface. If the physics interface is removed from the **Model Builder**, for example if **Moisture Transport in Solids** is deleted, then the **Moisture transport** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the **Solid mechanics** or **Moisture transport** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

POROELASTIC COUPLING PROPERTIES

The Biot theory is commonly used where the compressibility of the solid and/or fluid phase is important. Note that even if both constituents are modeled as incompressible,

the porous medium can still undergo large volume changes as a result of moisture inflow or outflow.

In the **Reference pressure level** list, choose **User defined** to enter a reference pressure p_{ref} (SI unit: Pa) at which the pore pressure does not affect the porous matrix deformation. The default value is `mts.pf_init`. If applicable, the list also contains options to inherit the value from another physics interface that announces the reference pressure level as a model input.

For the **Biot-Willis coefficient** list, the default is to take its value **From material**. Choose **User defined** to enter a Biot–Willis coefficient α_{B} (dimensionless) in the text field.

PORE PRESSURE

The multiphysics interface **Poroelasticity, Solid** couples the Solid Mechanics and the Darcy's Law interfaces, while the **Unsaturated Poroelasticity** interface couples the Solid Mechanics and the Moisture Transport in Solids interfaces. Although the theory for the **Poroelasticity, Solid** and **Unsaturated Poroelasticity** interfaces differs significantly as shown in Theory for the Poroelasticity Interfaces, the settings look similar. The main difference is how the pore pressure is computed.

For more information about the interface settings and the settings for the multiphysics node, see The Unsaturated Poroelasticity Interface and Unsaturated Poroelasticity (Multiphysics Coupling).

The Multiphysics interface **Unsaturated Poroelasticity**, is only available with some COMSOL products. For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

Overview of Electromagnetics–Structure Interaction Interfaces

The **Model Wizard** and the **Add Physics** window contain a number of multiphysics interfaces for modeling of structure interaction with magnetic and electric fields, all available under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ). The multiphysics interfaces are listed here and also described in more details in the subsequent sections.

- **Electromechanics** ( ) branch
  - The Electromechanics, Solid Interface ( )
  - The Electromechanics, Shell Interface ( )
  - The Electromechanics, Membrane Interface ( )
  - The Electromechanics, Boundary Elements Interface ( )
- **Piezoelectricity** ( ) branch
  - The Piezoelectricity, Solid Interface ( )
  - The Piezoelectricity, Layered Shell Interface ( )
- The Electrostriction Interface ( )
- The Ferroelectroelasticity Interface ( )
- **Magnetostriction** ( ) branch
  - The Piezomagnetism Interface ( )
  - The Nonlinear Magnetostriction Interface ( )
- **Magnetomechanics** ( ) branch
  - The Magnetomechanics, Solid Interface ( )
  - The Magnetomechanics, Shell Interface ( )
  - The Magnetomechanics, Membrane Interface ( )
  - The Magnetomechanics, No Currents, Solid Interface ( )
  - The Magnetomechanics, No Currents, Shell Interface( )
  - The Magnetomechanics, No Currents, Membrane Interface ( )
  - The Magnetic–Elastic Interaction in Rotating Machinery Interface ( )
  - The Magnetic–Rigid-Body Interaction in Rotating Machinery Interface ( )

- **Piezoresistivity** ( ) branch
  - The Piezoresistivity, Domain Currents Interface ( )
  - The Piezoresistivity, Boundary Currents Interface ( )
  - The Piezoresistivity, Shell Interface ( )

The Electromechanics, Solid Interface

The **Electromechanics, Solid** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Electrostatics with a moving mesh functionality to model the deformation of electrostatically actuated structures. The **Electromechanics, Solid** interface is available for 3D, planar 2D, and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and an **Electrostatics** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Electromechanics, Solid.

The full functionality of the Solid Mechanics and Electrostatics interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Electrostatics** interface by default adds a **Free Space** node and a **Charge Conservation in Solids** node with its selection set to all domains. This node serves to represent solid dielectric material domains. The **Free Space** node serves to represent dielectric domains adjacent to the solid. Any number of **Charge Conservation in Solids** or **Charge Conservation in Fluids** nodes can be added manually to the interface and configured to represent either solid or nonsolid materials.

Deforming Domain

When an **Electromechanics, Solid** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals). Domains selected in the Solid Mechanics interface are not applicable.

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Electromechanics, Shell Interface

The **Electromechanics, Shell** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Shell and Electrostatics with a moving mesh functionality to model the deformation of electrostatically actuated shells. The **Electromechanics, Shell** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Shell** interface and an **Electrostatics** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Electromechanics, Boundary.

The full functionality of the Shell and Electrostatics interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Electrostatics** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the shell domains. Any number of **Charge Conservation in Solids** or **Charge Conservation in Fluids** nodes can be added manually to the interface and configured to represent either solid or nonsolid materials.

Deforming Domain

When an **Electromechanics, Shell** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Electromechanics, Membrane Interface

The **Electromechanics, Membrane** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Membrane and Electrostatics with a moving mesh functionality to model the deformation of electrostatically actuated membranes. The **Electromechanics, Membrane** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Membrane** interface and an **Electrostatics** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Electromechanics, Boundary.

The full functionality of the Membrane and Electrostatics interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Electrostatics** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the membrane domains. Any number of **Charge Conservation in Solids** or **Charge Conservation in Fluids** nodes can be added manually to the interface and configured to represent either solid or nonsolid materials.

Deforming Domain

When an **Electromechanics, Membrane** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Electromechanics, Boundary Elements Interface

The **Electromechanics, Boundary Elements** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Electrostatics, Boundary Elements to model the deformation of electrostatically actuated structures. The **Electromechanics, Boundary Elements** interface is available for 3D and planar 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and an **Electrostatics, Boundary Elements** interface are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Electromechanics, Boundary.

The Piezoelectricity, Solid Interface

The **Piezoelectricity, Solid** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Electrostatics together with the constitutive relationships required to model piezoelectric phenomena. Both the direct and inverse piezoelectric effects can be modeled, and the piezoelectric coupling can be formulated using either the strain–charge or stress–charge forms.

When a predefined **Piezoelectricity, Solid** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or from **Add Physics** windows, **Solid Mechanics** and **Electrostatics** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature Piezoelectricity.

The participating **Solid Mechanics** interface includes the default **Piezoelectric Material** feature with its selection set to all domains. The **Electrostatics** interface has a default **Charge Conservation, Piezoelectric** feature with similar settings.

Such features can be also added manually to their corresponding interfaces similar to any other material model therein.

The Piezoelectricity multiphysics coupling node can be active only on the selection, where both features **Piezoelectric Material** and **Charge Conservation, Piezoelectric** are active.

You input both the mechanical and electrical material data under the Piezoelectric Material node. The data can be presented in either stress–charge or strain–charge form.

When it is used without an active Piezoelectricity coupling feature, the Piezoelectric Material node works similarly to a Linear Elastic Material feature with the material data input limited to anisotropic form using Voigt notations. All the electric material data has no effect.

You use the Charge Conservation, Piezoelectric feature under Electrostatics to select those domains, where the material is supposed to experience piezoelectric coupling. When used without a counterpart under the Solid Mechanics interface (or without the coupling feature), the Charge Conservation, Piezoelectric node acts as an ordinary

Charge Conservation feature with its material data input limited to the electric permittivity only.

All solid mechanics and electrostatics functionality for modeling is also accessible to include surrounding elastic solids or air domains. For example, add any solid mechanics material for other solid domain, a dielectric model for air (via Charge Conservation feature), or a combination. Note that in order to model a nonsolid dielectric domain, you need to remove such domain from the domain selection for the entire Solid Mechanics interface. This is because all material models under that interface represent solid materials (with the Linear Elastic Material node being always present and active in all those domains, where it is not explicitly overridden by any other material model).

In 2D and 2D axial symmetry, adding a Piezoelectricity interface also adds predefined base-vector coordinate systems for the material's (in the plane 2D case) *XY*-, *YZ*-, *ZX*-, *YX*-, *XZ*-, and *XY*-planes. These additional coordinate systems are useful for simplifying the material orientation for the piezoelectric material.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Electrostatics interface is described in The Electrostatics Interface in the *AC/DC Module User's Guide*.

In previous versions of COMSOL Multiphysics, a specific physics interface called **Piezoelectric Devices** was added to the **Model Builder**. Now, a predefined multiphysics coupling approach is used, improving the flexibility and design options for your modeling. For specific details, see The Multiphysics Branch and Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Piezoelectricity**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Solid Mechanics and Electrostatics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose the available coupling feature **Piezoelectricity** but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Feature

The Piezoelectricity coupling node is used to couple the interfaces.

Additional Features

Additional nodes and subnodes available with this multiphysics interface are described with the interfaces where they are available. The **Coupling Loss**, **Dielectric Loss**, **Mechanical Damping**, and **Conduction Loss (Time-Harmonic)** subnodes are available for Piezoelectric Material under the Solid Mechanics interface. The Charge Conservation, Piezoelectric node is described in the Electrostatics interface.

- Piezoelectricity
- Charge Conservation, Piezoelectric
- Modeling Piezoelectric Problems

Piezoelectric Shear-Actuated Beam: Application Library path **Structural_Mechanics_Module/Piezoelectric_Effects/shear_bender**

Surface Acoustic Wave Gas Sensor: Application Library path **MEMS_Module/Sensors/saw_gas_sensor**

Coupling Piezoelectricity with Acoustics

Using piezoelectric materials for an acoustic application is common, such as in sonars, microphones, sensors, and so forth. This is why coupling piezoelectricity with acoustic domains is of particular interest for these applications.

The multiphysics couplings between Solid Mechanics and the acoustics interfaces require the Acoustics Module.

Compared to a single piezoelectric model, you need to add a pressure acoustics physics interface, for example, **Pressure Acoustics, Frequency Domain** or **Pressure Acoustics, Transient** (depending on which study type you want to use) plus an **Acoustic-Structure Boundary** coupling under the **Multiphysics** node. You can also directly create the nodes that are needed for coupling by adding an Acoustic-Piezoelectric Interaction interface from the **Model Wizard** or **Add Physics** windows. If solid and acoustic domains are correctly defined, then the right coupling boundaries are automatically selected. Then specify domains of application for each physics.

- Select solid domains and **Piezoelectric Material** domains in **Solid Mechanics**.
- Select electrostatic domains and **Charge Conservation, Piezoelectric** domains in **Electrostatics**.
- Select acoustic domains in a **Pressure Acoustics** node.
- Under the **Multiphysics** branch, confirm that selections for **Piezoelectricity** and **Acoustic-Structure Boundary** are the right ones. If several **Pressure Acoustics**, **Solid Mechanics**, or **Electrostatics** interfaces are present, select the right ones that need to be coupled in the multiphysics interfaces.
- Continue the modeling process by entering the settings for each physics interface and feature and define materials.

The Piezoelectricity, Layered Shell Interface

The **Piezoelectricity, Layered Shell** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines structural layered shells and electric currents in shells together with the constitutive relationships required to model the direct and inverse piezoelectric effects. The piezoelectric coupling can be formulated using either the strain–charge or stress–charge forms.

When a predefined **Piezoelectricity, Layered Shell** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or from **Add Physics** windows, **Layered Shell** and **Electric Currents in Layered Shells** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature Piezoelectricity, Layered.

The **Layered Shell** interface includes a **Piezoelectric Material** feature. The **Electric Currents in Layered Shells** interface includes a **Piezoelectric Layer** feature with similar settings. Such features can be also added manually to their corresponding interfaces similar to any other material model therein.

The Piezoelectricity, Layered multiphysics coupling node can be active only on the selection, where both features **Piezoelectric Material** and **Piezoelectric Layer** are active.

You input both the mechanical and electrical material data under the Piezoelectric Material node. The data can be presented in either stress–charge or strain–charge form.

When it is used without an active Piezoelectricity, Layered coupling feature, the Piezoelectric Material node works similarly to a Linear Elastic Material feature with the material data input limited to anisotropic form using Voigt notations. All the electric material data has no effect.

You use the Piezoelectric Layer feature under Electric Currents in Shells to select those domains, where the material is supposed to experience piezoelectric coupling. When used without a counterpart under the Layered Shell interface (or without the coupling feature), the Piezoelectric Layer node acts as an ordinary Dielectric Layer feature with its material data input limited to the electric permittivity only.

All structural and electrostatics functionality for modeling is also accessible to include surrounding elastic solids or air domains. For example, add any s material model for other solid domain, a dielectric model for air, or a combination. Note that in order to model a nonsolid dielectric domain, you need to remove such domain from the domain selection for the entire Layered Shell interface. This is because all material models under that interface represent solid materials (with the Linear Elastic Material node being always present and active in all those domains, where it is not explicitly overridden by any other material model).

On the Constituent Physics Interfaces

- The Layered Shell interface is described in The Layered Shell Interface.
- The Electric Currents in Layered Shells interface is described in The Electric Currents, Layered Shell Interface in the *AC/DC Module User's Guide*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Piezoelectricity, Layered Shell**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Layered Shell and Electric Currents in Layered Shells interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can add the available coupling feature **Piezoelectricity, Layered** but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Feature

The Piezoelectricity, Layered coupling node is used to couple the interfaces.

Additional Features

Additional nodes and subnodes available with this multiphysics interface are described with the interfaces where they are available. The **Coupling Loss**, **Dielectric Loss** and **Mechanical Damping** subnodes are available for the Piezoelectric Material under the Layered Shell interface. The Piezoelectric Layer node is described for the Electric Currents in Layered Shells interface.

	• Piezoelectricity • Modeling Piezoelectric Problems

The Electrostriction Interface

The **Electrostriction** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Electrostatics together with the constitutive relationships required to model electrostriction in electrostatically actuated structures in regimes when the electric polarization can be assumed to vary linear with the applied electric field. Both the direct and inverse electrostrictive effects can be modeled.

The **Electrostriction** interface is available for 3D, planar 2D, and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and an **Electrostatics** interface are added to the **Model Builder**.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature Electrostriction.

All structural and electrostatics functionality for modeling is also accessible to include surrounding solids, fluids, or free space. For example, add any material model for other solid domains, a dielectric model for fluid, or a combination.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Electrostatic interface is described in The Electrostatics Interface in the *AC/DC Module User's Guide*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Electrostriction**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Solid Mechanics and Electrostatics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can add the available coupling feature **Electrostriction** but the modified settings are not included.

The **Electrostatics** interface by default adds the **Charge Conservation in Solids** node with its selection set to all domains. This node serves to represent solid dielectric material domains. Any number of **Charge Conservation in Solids** or **Charge Conservation in Fluids** nodes can be added to the interface.

- Electrostriction
- Modeling Electrostrictive and Ferroelectroelastic Materials

The Ferroelectroelasticity Interface

The **Ferroelectroelasticity** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Electrostatics together with the constitutive relationships required to model nonlinear electromechanical interaction in ferroelectric and piezoelectric materials. Electric polarization in such materials depends nonlinearly on the applied electric field including possible hysteresis and saturation effects. In addition, the polarization and mechanical deformations in such materials can be strongly coupled due to the electrostriction effect.

The **Ferroelectroelasticity** interface is available for 3D, planar 2D, and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and an **Electrostatics** interface are added to the **Model Builder**.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Electrostriction.

The participating **Electrostatics** interface includes the default **Charge Conservation, Ferroelectric** feature with its selection set to all domains. Such features can be also added manually to any **Electrostatics** interface in the model.

All structural and electrostatics functionality for modeling is also accessible to include surrounding solids, fluids, or free space. For example, add any material model for other solid domains, a dielectric model for fluid, or a combination.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Electrostatic interface is described in The Electrostatics Interface in the *AC/DC Module User's Guide*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Ferroelectroelasticity**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Solid Mechanics and Electrostatics interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can add the available coupling feature **Ferroelectroelasticity** but the modified settings are not included.

The **Electrostatics** interface by default adds the **Charge Conservation, Ferroelectric** node with its selection set to all domains. This node serves to represent solid ferroelectric material domains. Any number of **Charge Conservation in Solids** or **Charge Conservation in Fluids** nodes can be added to the interface.

- Ferroelectroelasticity
- Ferroelectricity
- Modeling Electrostrictive and Ferroelectroelastic Materials

Hysteresis in Piezoelectric Ceramics: Application Library path **MEMS_Module/Piezoelectric_Devices/piezoelectric_hysteresis**

The Magnetomechanics, Solid Interface

The **Magnetomechanics, Solid** ( ) interface, found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines **Solid Mechanics** and **Magnetic Fields** interfaces together with a moving mesh functionality. The interface can be used for modeling deformation of magnetically actuated structures, which includes interaction of magnetic fields with magnetic materials and current carrying elements such as coils and wires.

The **Magnetomechanics, Solid** interface is available for 3D, planar 2D, and axisymmetric 2D geometries.When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and a **Magnetic Fields** interface are added to the **Model Builder**.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Solid.

The full functionality of the Solid Mechanics and Magnetic Fields interfaces is accessible under the respective interface at the domain, boundary, edge, or point level in the geometry. The participating **Magnetic Fields** interface by default adds the **Ampère's Law in Solids** node. This node serves to represent solid magnetic material domains. Any number of **Ampère's Law** nodes can also be added manually to the interface to represent either solid or fluid materials.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Magnetic Fields interface is described in The Magnetic Fields Interface in the *AC/DC Module User's Guide*.

Deforming Domain

When a **Magnetomechanics, Solid** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid

dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

- Magnetomechanics
- Modeling Magnetostrictive Materials

The Magnetomechanics, Shell Interface

The **Magnetomechanics, Shell** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Shell and Magnetic Fields with a moving mesh functionality to model the deformation of magnetically actuated shells. The **Magnetomechanics, Shell** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Shell** interface and an **Magnetic Fields** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Boundary.

The full functionality of the Shell and Magnetic Fields interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Magnetic Fields** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the shell domains. Any number of **Ampère's Law in Solids** or **Ampère's Law in Fluids** nodes can be added manually to the interface to represent either solid or nonsolid materials.

Deforming Domain

When an **Magnetomechanics, Shell** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Yeoh**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Magnetomechanics, Membrane Interface

The **Magnetomechanics, Membrane** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Membrane and Magnetic Fields with a moving mesh functionality to model the deformation of magnetically actuated membranes. The **Magnetomechanics, Membrane** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Membrane** interface and an **Magnetic Fields** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Boundary.

The full functionality of the Membrane and Magnetic Fields interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Magnetic Fields** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the shell domains. Any number of **Ampère's Law in Solids** or **Ampère's Law in Fluids** nodes can be added manually to the interface to represent either solid or nonsolid materials.

Deforming Domain

When an **Magnetomechanics, Membrane** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Yeoh**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Magnetomechanics, No Currents, Solid Interface

The **Magnetomechanics, No Currents, Solid** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines **Solid Mechanics** and **Magnetic Fields, No Currents** interfaces with a moving mesh functionality to model the deformation of magnetostatically actuated structures. The interface is suitable in cases when induction and conduction effects such as eddy currents are negligible within the structure.

The **Magnetomechanics, No Currents, Solid** interface is available for 3D, planar 2D, and axisymmetric 2D geometries. When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and a **Magnetic Fields, No Currents** interface are added to the **Model Builder**.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Solid.

The full functionality of the Solid Mechanics and Magnetic Fields, No Currents interfaces is accessible under the respective interfaces at the domain, boundary, edge, or point level in the geometry. The participating **Magnetic Fields, No Currents** interface by default adds the **Magnetic Flux Conservation in Solids** node with its selection set to all domains. This node serves to represent solid magnetic material domains. Any number of **Magnetic Flux Conservation in Solids** or **Magnetic Flux Conservation in Fluids** nodes can be added manually to the interface to represent either solid or fluid materials.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Magnetic Fields, No Currents interface is described in The Magnetic Fields, No Currents Interface in the *AC/DC Module User's Guide*.

Deforming Domain

When a **Magnetomechanics, No Currents, Solid** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains

and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Hyperelastic**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

- Magnetomechanics
- Modeling Magnetostrictive Materials

The Magnetomechanics, No Currents, Shell Interface

The **Magnetomechanics, No Currents, Shell** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Shell and Magnetic Fields, No Currents with a moving mesh functionality to model the deformation of magnetostatically actuated shells. The **Magnetomechanics, No Currents, Shell** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Shell** interface and a **Magnetic Fields, No Currents** interface are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Boundary.

The full functionality of the Shell and Magnetic Fields, No Currents interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Magnetic Fields, No Currents** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the shell domains. Any number of **Magnetic Flux Conservation in Solids** or **Magnetic Flux Conservation in Fluids** nodes can be added manually to the interface to represent either solid or nonsolid materials.

Deforming Domain

When an **Magnetomechanics, No Currents, Shell** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Yeoh**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Magnetomechanics, No Currents, Membrane Interface

The **Magnetomechanics, No Currents, Membrane** interface ( ), found under the **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Membrane and Magnetic Fields, No Currents with a moving mesh functionality to model the deformation of magnetostatically actuated membranes. The **Magnetomechanics, No Currents, Membrane** interface is available for 3D and axisymmetric 2D geometries.

When this multiphysics interface is added using the **Model Wizard**, a **Membrane** interface and a **Magnetic Fields, No Currents** interfaces are added to the **Model Builder**. In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature Magnetomechanics, Boundary.

The full functionality of the Membrane and Magnetic Fields, No Currents interfaces is accessible under the corresponding interfaces at the domain, boundary line, or point level in the geometry. The participating **Magnetic Fields, No Currents** interface by default adds the **Free Space** node with its selection set to all domains. This node serves to represent dielectric domains adjacent to the shell domains. Any number of **Magnetic Flux Conservation in Solids** or **Magnetic Flux Conservation in Fluids** nodes can be added manually to the interface to represent either solid or nonsolid materials.

Deforming Domain

When an **Magnetomechanics, No Currents, Membrane** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** and a **Symmetry/Roller** nodes are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selection. Under the **Deforming Domain**, you can select those domain that represent nonsolid dielectric materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could just introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals).

Under the **Deforming Domain**, you can also choose the **Mesh smoothing type**, by default set to **Yeoh**. More information of the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Symmetry** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

The Magnetic–Elastic Interaction in Rotating Machinery Interface

The **Magnetic–Elastic Interaction in Rotating Machinery** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Rotating Machinery, Magnetic interfaces together with moving mesh features. The interface can be used for modeling deformation of rotating magnetic machinery structures such as electric motors and generators. Using this interface, you can, for example, compute how stresses are distributed in both stator and rotor of an electric motor.

The **Magnetic–Elastic Interaction in Rotating Machinery** interface is available for 3D, and planar 2D geometries. When this multiphysics interface is added using the **Model Wizard**, a **Solid Mechanics** interface and a **Rotating Machinery, Magnetic** interface are added to the **Model Builder**.

In addition, a Magnetic Forces, Rotating Machinery multiphysics coupling node is added under **Multiphysics Couplings**.

On the Constituent Physics Interfaces

- The Rotating Machinery, Magnetic interface is described in The Rotating Machinery, Magnetic Interface.
- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Deforming Domain feature is described in the Moving Mesh Features.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, all specific settings are included with the physics interfaces. However, if physics interfaces are added one at a time followed by the coupling features, these modified settings are not automatically included.

The participating **Rotating Machinery, Magnetic** interface by default adds an **Ampère's Law** node with its selection set to all domains, and the **Material type** set to **Solid**. This node serves to represent solid magnetic material domains. Any number of **Ampère's Law** nodes can be added manually to the interface and configured to represent either solid or nonsolid (for example, air or free space) materials.

The **Solid Mechanics** interface by default adds a **Rotating Frame** node to model the effect of rotation. To add the rotation to the definition of the spatial frame, the **Define spatial frame rotation** checkbox is selected by default.

MOVING MESH

When a **Magnetic–Elastic Interaction in Rotating Machinery** multiphysics interface is added using the **Model Wizard**, a **Deforming Domain** node and a **Rotating Boundary** node are added automatically in **Model Builder** under the **Definitions** node. These nodes serve to represent domains and boundaries where the mesh can deform. By default, both nodes have empty selections. Under the **Deforming Domain**, you can select domains that represent nonsolid magnetic materials such as air, fluid, and free space. However, this is only needed if the boundaries of such domains experience significant changes due to the deformation or rotation of the adjacent solid domains. Otherwise, the moving mesh computations could introduce unnecessary overhead and nonlinearity in the model (see Deformed Mesh Fundamentals). Domains selected in the Solid Mechanics are not applicable.

Under the **Deforming Domain** you can also choose the **Mesh smoothing type**, which is by default set to **Hyperelastic**. More information about the smoothing type can be found in the Deforming Domain chapter in the *COMSOL Multiphysics Reference Manual*.

By default, the mesh is fixed at all exterior boundaries of the geometry. Use the **Rotating Boundary** node to select the exterior boundaries, where the mesh can slide along the boundary. You can also add other types of boundary conditions for the mesh motion, for details see Moving Mesh Features in the *COMSOL Multiphysics Reference Manual*.

COUPLING FEATURES

Coupling Features

The Magnetic Forces, Rotating Machinery multiphysics coupling is used to couple the interfaces.

The Piezomagnetism Interface

The **Piezomagnetism** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Magnetic Fields together with the constitutive relationships required to model linear magnetostrictive materials and devices. Both the direct and inverse piezomagnetic effects can be modeled.

When a predefined **Piezomagnetism** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or from **Add Physics** windows, **Solid Mechanics** and **Magnetic Fields** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature Piezomagnetism.

The participating **Solid Mechanics** interface includes the default **Piezomagnetic Material** feature with its selection set to all domains. The **Magnetic Fields** interface gets a default **Ampère's Law, Piezomagnetic** feature with similar settings.

Such features can be also added manually to their corresponding interfaces similar to any other material models therein.

The Piezomagnetism multiphysics coupling node can be active only on the selection where the features Piezomagnetic Material and Ampère's Law, Piezomagnetic are both active.

Input the mechanical and magnetic material data under the Piezomagnetic Material, and the electrical properties under Ampère's Law, Piezomagnetic.

When used without an active Piezomagnetism coupling feature, the Piezomagnetic Material node works similarly to a Linear Elastic Material feature with some limitations on the format for the elastic material data input. The magnetic material data and coupling data will have no effect.

Use the Ampère's Law, Piezomagnetic feature under Magnetic Fields to select the domains where the material is supposed to experience linear magnetostrictive coupling. When used without a counterpart under the Solid Mechanics interface (and/or without the coupling feature) Ampère's Law, Piezomagnetic node acts as an ordinary Ampère's Law feature with its material data input limited to the electric properties only. The magnetic permittivity of free space will be assumed.

All solid mechanics and magnetics functionalities for modeling are also accessible to include surrounding elastic solids or air domains. For example, add any solid mechanics material for other solid domain, a nonsolid model for air (via Ampère's Law feature), or a combination. Note that in order to model a nonsolid magnetic material, you need to remove such domain from the domain selection for the entire Solid Mechanics interface. This is because all material models under that interface represent solid materials (with the Linear Elastic Material node being always present and active in all those domains, where it is not explicitly overridden by any other material model).

In 2D and 2D axial symmetry, adding a Piezomagnetism interface also adds predefined base-vector coordinate systems for the material's (in the plane 2D case) *XY*-, *YZ*-, *ZX*-, *YX*-, *XZ*-, and *XY*-planes. These additional coordinate systems are useful for simplifying the material orientation for nonisotropic materials.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Magnetic Fields interface is described in The Magnetic Fields Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined multiphysics interface, for example **Piezomagnetism**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Solid Mechanics and Magnetic Fields interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose the available coupling feature **Piezomagnetism** but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Feature

The Piezomagnetism coupling node is used to couple the interfaces.

Additional Features

Additional nodes and subnodes available with this multiphysics interface are described with the interfaces where they are available. The **Mechanical Damping** and **Thermal Expansion** subnodes are available for Piezomagnetic Material under the Solid Mechanics interface. The Ampère's Law, Piezomagnetic node is described for the Magnetic Fields interface.

- Magnetostriction and Piezomagnetism
- Ampère's Law, Piezomagnetic
- Modeling Magnetostrictive Materials

The Nonlinear Magnetostriction Interface

The **Nonlinear Magnetostriction** interface ( ), found under **Electromagnetics–Structure Interaction** ( ) branch within the **Structural Mechanics** branch ( ) or the **Electromagnetics and Mechanics** ( ) branch within the **AC/DC** branch ( ) when adding a physics interface, combines Solid Mechanics and Magnetic Fields together with the constitutive relationships required to model magnetostrictive materials and devices. Both the direct and inverse magnetostriction effects can be modeled.

When a predefined **Magnetostriction** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or from **Add Physics** windows, **Solid Mechanics** and **Magnetic Fields** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature Nonlinear Magnetostriction.

The participating **Solid Mechanics** interface will use the default **Linear Elastic Material** feature with its selection set to all domains. The **Magnetic Fields** interface gets a default **Ampère's Law, Nonlinear Magnetostrictive** feature with similar settings.

Such features can be also added manually to their corresponding interfaces similar to any other material models therein.

Input the mechanical material data in the corresponding node under Solid Mechanics, the magnetic and electrical properties under Ampère's Law, Nonlinear Magnetostrictive, and the coupling data in the Nonlinear Magnetostriction multiphysics coupling node.

Use the Ampère's Law, Nonlinear Magnetostrictive feature under Magnetic Fields to select the domains where the material is supposed to experience magnetostrictive coupling. When used without a counterpart under the Solid Mechanics interface (and/or without the coupling feature) Ampère's Law, Nonlinear Magnetostrictive node acts as an ordinary Ampère's Law feature set to use Analytic Magnetization Curve. The coupling data will have no effect.

All solid mechanics and magnetics functionalities for modeling are also accessible to include surrounding elastic solids or air domains. For example, add any solid mechanics material for other solid domain, a nonsolid model for air (via Ampère's Law feature), or a combination. Note that in order to model a nonsolid magnetic material,

you need to remove such domain from the domain selection for the entire Solid Mechanics interface. This is because all material models under that interface represent solid materials (with the Linear Elastic Material node being always present and active in all those domains, where it is not explicitly overridden by any other material model).

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Magnetic Fields interface is described in The Magnetic Fields Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined multiphysics interfaces such as **Nonlinear Magnetostriction**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if both Solid Mechanics and Magnetic Fields interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose the available coupling feature **Nonlinear Magnetostriction** but the modified settings are not included.

PHYSICS INTERFACES AND COUPLING FEATURES

Coupling Feature

The Nonlinear Magnetostriction coupling node is used to couple the interfaces.

Additional Features

Additional nodes and subnodes available with this multiphysics interface are described with the interfaces where they are available. The **Mechanical Damping** and **Thermal Expansion** subnodes are available for the Magnetostrictive Material under the Solid Mechanics interface. The Ampère's Law, Nonlinear Magnetostrictive node is described for the Magnetic Fields interface.

- Magnetostriction and Piezomagnetism
- Ampère's Law, Nonlinear Magnetostrictive
- Modeling Magnetostrictive Materials

	Nonlinear Magnetostrictive Transducer: Application Library path **Structural_Mechanics_Module/Magnetomechanics/ nonlinear_magnetostriction**

The Phase-Field Damage Interface

The **Phase-Field Damage** multiphysics interface ( ) combines the Solid Mechanics and Phase Field in Solids interfaces to model the evolution of damage and cracks in deforming solids.

The coupling occurs on the domain level, where the driving force for damage depends on the state of stress and deformation in the solid, which governs the evolution of the phase field. In turn, the phase field determines the degradation of the solid properties in the damaged regions.

The interface can be used for stationary and time-dependent analyses in 3D, 2D, and 2D axisymmetric domains.

When a predefined **Phase-Field Damage** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, the **Solid Mechanics** and **Phase Field in Solids** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the **Phase-Field Damage** multiphysics coupling.

On the Constituent Physics Interfaces

- The Solid Mechanics interface is described in The Solid Mechanics Interface.
- The Phase Field in Solids interface is described in The Phase Field in Solids Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Phase-Field Damage**, specific settings are included with the physics interfaces and the coupling features.

However, if the physics interfaces are added one at a time, followed by the coupling features, these predefined settings are not automatically included.

For example, if single **Solid Mechanics** and **Phase Field in Solids** interfaces are added, an empty **Multiphysics** node appears in the model tree. You can choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-11: MODIFIED SETTINGS FOR THE PHASE-FIELD DAMAGE INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior, the **Structural transient behavior** is set to **Quasistatic**.
Phase Field in Solids	For the Phase Field in Solids interface, a **Phase Field Model** node is added by default with the **AT2** phase field model.
Phase-Field Damage	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Phase Field in Solids interfaces are preselected in the Coupled Interfaces section.

The Phase-Field Damage, Explicit Dynamics Interface

The **Phase-Field Damage, Explicit Dynamics** multiphysics interface ( ) combines the Solid Mechanics, Explicit Dynamics and Phase Field in Solids interfaces to model the evolution of damage and cracks in deforming solids.

The coupling occurs on the domain level, where the driving force for damage depends on the state of stress and deformation in the solid, which governs the evolution of the phase field. In turn, the phase field determines the degradation of the solid properties in the damaged regions.

The interface can be used for stationary and time-dependent analyses in 3D, 2D, and 2D axisymmetric domains.

When a predefined **Phase-Field Damage** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, the **Solid Mechanics, Explicit Dynamics** and **Phase Field in Solids** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the **Phase-Field Damage** multiphysics coupling.

On the Constituent Physics Interfaces

- The Solid Mechanics, Explicit Dynamics interface is described in The Solid Mechanics, Explicit Dynamics Interface.
- The Phase Field in Solids interface is described in The Phase Field in Solids Interface.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Phase-Field Damage**, specific settings are included with the physics interfaces and the coupling features.

However, if the physics interfaces are added one at a time, followed by the coupling features, these predefined settings are not automatically included.

For example, if single **Solid Mechanics, Explicit Dynamics** and **Phase Field in Solids** interfaces are added, an empty **Multiphysics** node appears in the model tree. You can

choose from the available coupling features, but the settings in the constituent interfaces are not modified.

TABLE 17-12: MODIFIED SETTINGS FOR THE PHASE-FIELD DAMAGE INTERFACE.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics, Explicit Dynamics	For the Solid Mechanics, Explicit Dynamics interface, under Structural Transient Behavior, the **Structural transient behavior** is set to **Include inertial terms**.
Phase Field in Solids	For the Phase Field in Solids interface, a **Phase Field Model** node is added by default with the **AT2** phase field model.
Phase-Field Damage	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding Solid Mechanics, Explicit Dynamics and Phase Field in Solids interfaces are preselected in the Coupled Interfaces section.

Multiphysics Couplings

Thermal Expansion

The **Thermal Expansion** coupling is similar to the Thermal Expansion (for Materials) node ( ) that can be added under, for example, the **Linear Elastic Material** or **Hyperelastic Material** for the Solid Mechanics interface. The purpose is the same, and if both nodes are used for the same selection, the settings in the coupling node takes precedence.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Thermal Expansion 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `tel`.

DOMAIN SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains to define the coefficient of thermal expansion and the different temperatures that cause thermal stress, or select **All domains** as needed.

When **Thermal Expansion** is added, the selection is the same as for the participating physics interfaces. Only domains that are active in the physics interfaces selected in the Coupled Interfaces section can be chosen.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Heat transfer** and **Solid mechanics** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a multiphysics interface is selected in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Thermal Expansion** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Heat Transfer in Solids** is deleted, then the **Heat transfer** list defaults to **None** as there is nothing to couple to.

The computed thermal strains are automatically made consistent with the structural strain. Hence, any discretization order can be used in the Heat Transfer in Solids interface.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the **Heat transfer** or **Solid mechanics** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual*.

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

where α is the secant coefficient of thermal expansion. α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - I$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

HEAT SOURCES

Select **Mechanical Losses** to make any energy dissipation computed in the structural interface act as a heat source in the heat transfer interfaces. Examples of such contributions are work done by plastic deformation or viscous damping.

Select **Thermoelastic damping** to include the reverse coupling where the changes in stress act as a heat source in the heat transfer analysis. Thermoelastic damping is only used when Structural Transient Behavior is set to **Include inertial terms**.

You can select to **Use volume reference temperature as linearization point** when there is no preceding stationary solution to determine the linearization point for a consequent frequency domain analysis.

	You need to select **Calculate dissipated energy** in the **Energy Dissipation** section of the settings for the material model in Solid Mechanics to add the variables for energy dissipation. To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

	• Temperature-Dependent Material Data • Thermoelastic Damping • Thermal Expansion and Thermoelastic Damping for large deformation formulation. • Entropy and Thermoelasticity

Thermal Expansion, Thin Layer

Use the **Thermal Expansion, Thin Layer** multiphysics coupling ( ) to add an internal thermal strain caused by changes in temperature and to account for the corresponding mechanical losses in the heat balance in thin layers represented by boundaries.

The **Thermal Expansion, Thin Layer** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Thermal Expansion, Thin Layer 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `tetl1`.

BOUNDARY SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific boundaries to define the coefficient of thermal expansion and the different temperatures that cause thermal stress, or select **All boundaries** as needed.

Only boundaries that are active in the physics interfaces selected in the **Coupled Interfaces** section can be selected.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Heat transfer** and **Structure** lists include all applicable physics interfaces. The first physics interface of each type in the component is selected as the default.

You can also select **None** from either list to uncouple the **Thermal Expansion, Thin Layer** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Heat Transfer in Solids** is deleted, then the **Heat transfer** list defaults to **None** as there is nothing to couple to.

BOUNDARY PROPERTIES

Select a **Thin layer type** — **Nonlayered** or **Layered**.

For **Layered** boundaries, select the applicable layers (the default setting is **Use all layers**) defining the required material properties for the node.

If no layered materials have been included yet, there is a shorthand available for creating a **Single Layer Material**, a **Layered Material Link**, or a **Layered Material Stack** (the plus symbol next to the **Layer** list). See Shell Properties for details.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{th} = \alpha(T - T_{ref})$$

where α is the secant coefficient of thermal expansion which can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

HEAT SOURCES

Select **Mechanical losses** to make any energy dissipation computed in the thin layer act as a heat source in the heat transfer interface. Examples of such contributions are work done by plastic deformation or viscous damping.

Select **Thermoelastic damping** to include the reverse coupling where the changes in stress act as a heat source in the heat transfer analysis. Thermoelastic damping is only used when Structural Transient Behavior is set to **Include inertial terms**.

You need to select **Calculate dissipated energy** in the **Energy Dissipation** section of the settings for the material model in Solid Mechanics to add the variables for energy dissipation. To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

- Temperature-Dependent Material Data
- Thermoelastic Damping
- Entropy and Thermoelasticity

Thermal Expansion, Layered

Use the **Thermal Expansion, Layered** multiphysics coupling ( ) to add an internal thermal strain caused by changes in temperature and account for the corresponding mechanical losses in the heat balance in layered materials represented by boundaries.

The **Thermal Expansion, Layered** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Thermal Expansion, Layered 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `tel1`.

BOUNDARY SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific boundaries to define the coefficient of thermal expansion and the different temperatures that cause thermal stress, or select **All boundaries** as needed.

Only boundaries that are active in the physics interfaces selected in the **Coupled Interfaces** section can be selected.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Heat transfer** and **Structure** lists include all applicable physics interfaces. The first physics interface of each type in the component is selected as the default.

You can also select **None** from either list to uncouple the **Thermal Expansion, Layered** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Heat Transfer in Shells** is deleted, then the **Heat transfer** list defaults to **None** as there is nothing to couple to.

Thermal strains are proportional to the temperature, while structural strains are computed from the gradient of the displacement field. It is a good practice to match the discretization order of thermal and structural strains.

When manually adding a **Thermal Expansion, Layered** multiphysics coupling, use a discretization one order lower for the temperature field than what is used for the displacement field.

This is automatically taken care of when you use The Thermal Stress, Shell Interface, The Thermal Stress, Layered Shell Interface, or The Thermal Stress, Membrane Interface.

SHELL PROPERTIES

Select the applicable layers (the default setting is **Use all layers**) defining the required material properties for the node.

If no layered materials have been included yet, there is a shorthand available for creating a **Single Layer Material**, a **Layered Material Link**, or a **Layered Material Stack** (the plus symbol next to the **Layer** list).

When a layered material stack or link is selected from the **Layer** list, clear the checkboxes corresponding to layers where the node should not be applied in the **Selection** table.

You can visualize the selected layered materials and layers in each layered material by clicking the **Layer cross section preview** and **Layer 3D preview** buttons.

See The Shell Properties and Interface Selection Sections.

The desired selection for the node may correspond to boundaries with different layered materials. The **All layered materials** option allows to gather these materials to make the desired selection applicable for the node on the union of the boundaries where the layered materials are defined.

See Layered Material, Layered Material Link, Layered Material Stack, Layered Material Link (Subnode), and Single Layer Materials in the *COMSOL Multiphysics Reference Manual* for details on the definition of layered materials.

See Thermal Expansion for a description of the corresponding multiphysics coupling in domains.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

See also

- Using Common Model Input
- Default Model Inputs and Model Input in the *COMSOL Multiphysics Reference Manual.*

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\varepsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

where α is the secant coefficient of thermal expansion, which can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{th} = \exp\left(\int_{T_{ref}}^{T} \alpha_t(\tau)d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

HEAT SOURCES

Select **Mechanical losses** to make any energy dissipation computed in the structural interface act as a heat source in the heat transfer interfaces. Examples of such contributions are work done by plastic deformation or viscous damping.

Select **Thermoelastic damping** to include the reverse coupling where the changes in stress act as a heat source in the heat transfer analysis. Thermoelastic damping is only used when Structural Transient Behavior is set to **Include inertial terms**.

You need to select **Calculate dissipated energy** in the **Energy Dissipation** section of the settings for the material model in Layered Shell to add the variables for energy dissipation. To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

- Temperature-Dependent Material Data
- Thermoelastic Damping
- Entropy and Thermoelasticity

Hygroscopic Swelling

The **Hygroscopic Swelling** multiphysics coupling node ( ) is for moisture concentration coupling between the Solid Mechanics interface and either the Transport of Diluted Species or Transport of Diluted Species in Porous Media interfaces.

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Hygroscopic Swelling 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `hs1`.

DOMAIN SELECTION

The domain selection is empty by default. Only domains that have either **Fluid**, **Solid**, or **Porous Medium** selected in the **Transport of Diluted Species** interface; and either **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** selected in the **Solid Mechanics** interface are available.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Moisture concentration** and **Structure** lists include all applicable physics interfaces.

You can select **None** from either list to uncouple the coupling node from a physics interface. If the physics interface is removed from the **Model Builder** (for example, **Solid Mechanics** is deleted), then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

HYGROSCOPIC SWELLING PROPERTIES

Enter a **Strain reference concentration** $c_{\mathrm{mo,ref}}$. This is the concentration at which there are no strains due to hygroscopic swelling.

Enter the **Molar mass** of the fluid, M_{m}. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** from the list to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h}.The default value for the **User defined** case is `1.5e-4` m^3/kg. When an anisotropic coefficient of hygroscopic swelling is used, the axis orientations are given by the coordinate system selection in the structural mechanics material node to which it contributes.

The **Include moisture as added mass** checkbox is selected by default. When selected, the mass of the fluid is included in a dynamic analysis, and when using mass proportional loads. It will also contribute when computing mass properties.

Shrinkage and Swelling

The **Shrinkage and Swelling** multiphysics coupling node ( ) is used to bidirectionally couple the **Solid Mechanics** interface and the **Transport in Solids** interface.

INFLUENCE OF TRANSPORT IN SOLIDS ON SOLID MECHANICS

The volumetric strain caused by the species transported in the solid reads

$$\varepsilon_{\mathrm{c,vol}} = \Omega_{\mathrm{Vc}}(c - c_{\mathrm{ref}})$$

where Ω_{Vc} is the volumetric coefficient of swelling, c is the species concentration, and c_{ref} is the strain-free reference concentration.

For a large strain formulation, the volumetric change caused by the transported species in the solid reads

$$J_{\mathrm{c}} = 1 + \Omega_{\mathrm{Vc}}(c - c_{\mathrm{ref}})$$

When the linear coefficient of swelling is selected, the strain resulting from swelling and shrinkage reads

$$\varepsilon_{\mathrm{c}} = \Omega_{\mathrm{c}}(c - c_{\mathrm{ref}})$$

and for a large strain formulation, the deformation gradient reads

$$F_c = I + \Omega_c(c - c_{ref})$$

where the linear coefficient of swelling, Ω_c, is an isotropic or diagonal tensor.

INFLUENCE OF SOLID MECHANICS ON TRANSPORT IN SOLIDS

The deformation caused by **Shrinkage and Swelling** node contributes to the flux within the **Transport in Solids** interface

$$\Gamma_{sas} = -M\nabla\mu_{el}$$

where M denotes the mobility, and μ_{el} denotes the derivative of the elastic contribution to the free energy, W_s, with respect to the transported species

$$\mu_{el} = \frac{\partial W_s}{\partial c}$$

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Shrinkage and Swelling 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `sas1`.

DOMAIN SELECTION

The domain selection is empty by default. Only domains that have **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** selected in the **Solid Mechanics** interface are available.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Transported quantity** and **Solid mechanics** lists include all applicable physics interfaces.

You can select **None** from either list to uncouple the coupling node from a physics interface. If the physics interface is removed from the **Model Builder** (for example, **Solid Mechanics** is deleted), then the list defaults to **None** as there is nothing to couple to.

SHRINKAGE AND SWELLING PROPERTIES

Select how the swelling coefficient is entered — **Volumetric coefficient** or **Linear coefficient** — from the **Specify** list.

Enter a **Reference value** c_{ref}. This is the concentration at which there are no strains due to shrinkage or swelling.

When **Volumetric coefficient** is selected, enter the volumetric **Swelling coefficient** Ω_{Vc}.

When **Linear coefficient** is selected, select **Isotropic** or **Diagonal** from the list to enter one or more components for the swelling tensor Ω_{c}. For anisotropic swelling, the axis orientations are given by the coordinate system selection in the Solid Mechanics node to which it contributes.

Enter the **Molar mass** M_{c}.

The units for the input depend on the settings in the Transported Quantity section.

Select the **Include mass added by transported quantities** checkbox to include the mass of the transported species in dynamic analyses, and when using mass proportional loads. The additional mass from the species will also be included when calculating the mass properties.

Piezoelectricity

The **Piezoelectricity** multiphysics coupling node ( ) passes the appropriate relative permittivity from the **Piezoelectric Material** node in the **Solid Mechanics** interface (where it is specified together with the other material properties of the piezoelectric) to the **Charge Conservation, Piezoelectric** node in the **Electrostatics** interface. The Charge Conservation, Piezoelectric node implements the domain level electrostatics equations and does not require any user settings when it is coupled with the multiphysics coupling node.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Piezoelectricity 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `pze1`.

DOMAIN SELECTION

The domain selection is locked so that all applicable domains are selected. Only domains that have both **Charge Conservation, Piezoelectric** selected in the **Electrostatics** interface and **Piezoelectric Material** selected in the **Solid Mechanics** interface are selected.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Solid mechanics** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a Piezoelectricity interface is selected in the **Model Wizard** or **Add Physics** window, then the participating Solid Mechanics and Electrostatics interfaces are selected.

You can also select **None** from either list to uncouple the Piezoelectricity node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Piezoelectricity, Layered

The **Piezoelectricity, Layered** multiphysics node ( ) passes the appropriate relative permittivity from the **Piezoelectric Material** node in the **Layered Shell** interface (where it is specified together with the other material properties of the piezoelectric) to the **Piezoelectric Layer** node in the **Electric Currents in Layered Shells** interface. The Piezoelectric Layer node implements the equations for the charge balance in shells, and

it does not require any user settings when the interface is coupled through the multiphysics coupling node.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Piezoelectricity, Layered 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `pze1`.

BOUNDARY SELECTION

The boundary selection is locked so that all applicable boundaries are selected. Only boundaries that have both **Piezoelectric Layer** selected in the **Electric Currents in Layered Shells** interface and **Piezoelectric Material** selected in the **Layered Shell** interface are selected.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Layered shell** and **Electric Currents in Layered Shells** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a Piezoelectricity, Layered Shell interface is selected in the **Model Wizard** or **Add Physics** window, then the participating Layered Shell and Electric Currents in Layered Shells interfaces are selected.

You can also select **None** from either list to uncouple the Piezoelectricity, Layered node from a physics interface. If the physics interface is removed from the **Model Builder**, for

example **Layered Shell** is deleted, then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Electrostriction

The **Electrostriction** multiphysics node ( ) passes the electric polarization contribution to strain from the **Electrostatics** interface to the **Solid Mechanics** interface. In contrast to linear piezoelectricity, the electrostrictive strain is quadratic in polarization.

It also passes the mechanical stress contribution to polarization from the **Solid Mechanics** interface to the **Electrostatics** interface, which is called the inverse electrostrictive effect.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Electrostriction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `efe1`.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Solid mechanics** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when either Electrostriction or Ferroelectroelasticity multiphysics interface has been selected in the **Model Wizard** or **Add Physics** window, then the participating Solid Mechanics and Electrostatics interfaces are selected.

You can also select **None** from either list to uncouple the Electrostriction node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual.*

DOMAIN SELECTION

The domain selection is set by default to all domains, so that all applicable domains are selected automatically. Such domains represent an intersection of the applicable domains under the corresponding **Electrostatics** and **Solid Mechanics** interfaces selected in the coupling feature.

In **Electrostatics** interface, the following two domains are applicable:

- **Charge Conservation in Solids**. Use this domain feature for solid dielectric materials, for which a linear dependency can be assumed for the electric polarization with respect to the applied electric field.
- **Charge Conservation, Ferroelectric**. Use this domain feature for solid ferroelectric or nonlinear piezoelectric materials.

In the **Solid Mechanics** interface, the following domain material features are applicable:

- **Linear Elastic Material**

- **Nonlinear Elastic Material** (with the Nonlinear Structural Materials Module)
- **Hyperelastic Material** (with the Nonlinear Structural Materials Module)

For nonsolid dielectric domains, remove them from the **Solid Mechanics** interface selection, and use the **Free Space** or **Charge Conservation in Fluids** nodes in the **Electrostatics** interface.

COUPLING TYPE

From the list, choose one of these coupling types:

- **Polarization contribution to strain** (the default) also known as direct electrostrictive effect, include only the deformation of the material caused by its polarization in response to the applied electric field.
- **Stress contribution to polarization**, also known as inverse electrostrictive effect, to include only the change in the material polarization as a result of applied mechanical stress or strain.
- **Fully coupled** to include both the direct and inverse electrostrictive effects.

ELECTROSTRICTION

From the list, choose one of these forms for the electrostrictive strain tensor:

- **Quadratic** (default)
- **Quadratic, deviatoric**. Use this form if the solid deformation due to the electrostrictive effect is volume preserving.
- **User defined**. Any valid expression can be entered for the electrostrictive strain tensor components. It is recommended to use coupling feature scoped variables for the polarization components, for example: `efe1.PX`.

When the **Electrostrictive strain tensor** is set to **Quadratic**, one of the following options can be selected from the **Material symmetry** list:

- **Isotropic** (default)
- **Cubic crystal**
- **Anisotropic**

For the **Anisotropic** symmetry, a component of a 6-by-6 symmetric electrostrictive coupling matrix can be entered using either the **Voigt** (default) or the **Standard** option for the **Material data ordering**.

For the **Cubic crystal** symmetry, only three independent components of the electrostrictive coupling matrix need to be entered.

For the **Isotropic** symmetry, you can enter two independent components of the electrostrictive coupling matrix using either Q-matrix (default) or M-matrix notations. Alternatively, you can enter values for the **Electrostrictive constants** a_1 and a_2 (SI units: F/m) using the definitions according to either Ref. 1 or Ref. 2.

You can neglect the terms quadratic in the reversible polarization components in the expression for the electrostrictive strain by checking the checkbox **Small reversible polarization**. This functionality is applicable only when this coupling node is used as a part of either Ferroelectroelasticity or Electrostriction multiphysics interface, and the corresponding **Electrostatics** interface is configured for hysteresis modeling.

When the **Electrostrictive strain tensor** is set to **Quadratic, deviatoric**, the material is assumed to be isotropic, and you can enter the saturation electrostriction and polarization.

If this coupling node is used as a part of Ferroelectroelasticity multiphysics interface, the **Charge conservation, Ferroelectric** feature under the corresponding **Electrostatics** interface also contains an input for the saturation polarization. That input will then become inaccessible (grayed out), and the value entered at this coupling node will be used in all computations.

REFERENCES

1. J.A. Stratton, *Electromagnetic Theory*, Cambridge, MA, 1941.

2. L.D. Landau and E.M. Lifshitz, *Electrodynamics of Continuous Media*, Pergamon Press, pp. 69–73, 1960.

Electromechanics, Solid

Use an **Electromechanics, Solid** ( ) coupling to model electromechanical interaction between deformable solids and electric fields, particularly in case when the deformations of the structure can significantly affect the electric field distribution.

The presence of such multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `eme1`.

DOMAIN SELECTION

The domain selection is locked, so that all applicable domains are selected automatically. Such domains represent an intersection of the applicable domains under the corresponding **Electrostatics** and **Solid Mechanics** interfaces selected in the coupling feature.

In **Electrostatics** interface, the following domains are applicable:

- **Charge Conservation in Solids**. Use this domain feature for solid dielectric materials.
- **Charge Conservation, Ferroelectric**
- **Terminal**. You can use the domain terminal feature to represent conductive solid materials such as, for example, doped silicon.

In **Solid Mechanics** interface, the following domain material feature are applicable:

- **Linear Elastic Material**
- **Rigid Material**
- **Nonlinear Elastic Material** (with the Nonlinear Structural Materials Module)
- **Hyperelastic Material** (with the Nonlinear Structural Materials Module).

For nonsolid dielectric domains, remove them from the **Solid Mechanics** interface selection but keep them selected in the **Electrostatics** interface selection.

ELECTROMAGNETIC STRESS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Here, you can select the **Electromagnetic Stress Tensor** to be used in the coupling. The following options are available:

- **Minkowski** (default)
- **Einstein-Laub**
- **Chu**

See the Electromechanics chapter for details.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Solid mechanics** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the **Solid mechanics** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Electromechanics, Boundary

Use an **Electromechanics, Boundary** ( ) coupling to model electromechanical interaction between deformable shells or membranes and electric fields, particularly in

case when the deformations of the thin structure can significantly affect the electric field distribution.

The presence of such multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `emfb1`.

BOUNDARY SELECTION

All applicable boundaries are selected automatically. Such domains represent an intersection of the applicable boundaries the corresponding **Electrostatics** and **Shell** or **Membrane** interfaces selected in the coupling feature.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Thin structure** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example

Membrane is deleted, then the **Thin structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Electromechanics, Boundary Elements

Use an **Electromechanics, Boundary Elements** ( ) boundary multiphysics coupling to model electromechanical interaction between deformable solid domains and nonsolid voids, where the electric field is computed using the boundary element method. The backward coupling to Electrostatics is due to the deformations of the boundaries.

The presence of such multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `emeb1`.

BOUNDARY SELECTION

Applicable boundaries are those that are both exterior boundaries to the domains selected under the **Solid Mechanics** interface and boundaries to the domains selected under the **Electrostatics, Boundary Elements** interface. In particular, all boundaries

external to **Solid Mechanics** selection are applicable if the selection of the **Electrostatics, Boundary Elements** contains **Infinite void**.

For nonsolid dielectric domains, remove them from the **Solid Mechanics** interface selection but keep them selected in the **Electrostatics, Boundary Elements** interface selection.

The coupling is a boundary load on a structure caused by the Maxwell stress at the selected boundaries. The backward coupling to Electrostatics is due to the change of the spatial position of the boundaries caused by the deformation of the structure.

Only electrically linear and isotropic materials can be modeled using **Electrostatics, Boundary Elements**. For more details, see The Electrostatics, Boundary Elements Interface chapter in the *AC/DC Module User's Guide*.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Solid mechanics** and **Electrostatics, Boundary Elements** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid**

Mechanics is deleted, then the **Solid mechanics** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Magnetomechanics, Solid

Use the **Magnetomechanics, Solid** ( ) coupling to model interaction between deformable solids and magnetic fields. It can be used to model cases with or without induction currents, depending on which magnetic field physics interface is used. The supported magnetic field interfaces are **Magnetic Fields** and **Magnetic Fields, No Currents**, which is coupled with the **Solid Mechanics** interface. The coupling can also handle the case when the deformation of the structure significantly affect the magnetic field distribution.

The presence of such a multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes, unless the **Only use Lorentz force** checkbox is selected.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Magnetomechanics, Solid 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `mmcpl1`.

DOMAIN SELECTION

The domain selection includes by default all applicable domains. Such domains represent an intersection of the applicable domains under the corresponding **Magnetic Fields** and **Solid Mechanics** interfaces selected in the coupling feature.

If the **Only use Lorentz force** checkbox is not selected, the following domain features are applicable in the **Magnetic Fields** and **Magnetic Fields, No Currents** interfaces:

- **Ampère's Law in Solids**. Use this domain feature for solid magnetic materials.
- **Ampère's Law, Nonlinear Magnetostrictive**
- **Magnetic Flux Conservation**, if its material type input is set to **Solid**. Use this domain feature for solid magnetic materials.
- **Coil**, if its material type input is set to **Solid**. You can use this feature to represent current carrying solid domains such as coils and wires.
- **Magnet**
- **Passive Conductor**

In the **Solid Mechanics** interface, the following domain material features are applicable:

- **Linear Elastic Material**
- **Nonlinear Magnetostrictive Material**
- **Nonlinear Elastic Material** (with the Nonlinear Structural Materials Module)
- **Hyperelastic Material** (with the Nonlinear Structural Materials Module)
- **Rigid Material**

- For nonsolid magnetic domains, remove them from the **Solid Mechanics** interface selection but keep them selected in the **Magnetic Fields** or **Magnetic Fields, No Currents** interface selection.
- **Ampère's Law, Nonlinear Magnetostrictive** is a part of The Nonlinear Magnetostriction Interface. The corresponding multiphysics coupling feature, Nonlinear Magnetostriction, can be combined with **Magnetomechanics**.

See also

- Magnetomechanics

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Solid mechanics** and **Magnetic Fields** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the **Solid mechanics** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

ELECTROMAGNETIC STRESS

This section is available only if one of the coupled interfaces is **Magnetic Fields, No Currents**.

Here, you can select the **Electromagnetic Stress Tensor** to be used in the coupling. The following options are available:

- **Minkowski** (default)
- **Einstein-Laub**
- **Chu**

See the Magnetomechanics chapter for details.

LORENTZ COUPLING

This section is available only if one of the coupled interfaces is **Magnetic Fields**.

Select the **Only use Lorentz force** checkbox in order to model the interaction by only applying a Lorentz force to the mechanics side and a Lorentz velocity term to the electromagnetics side. The Lorentz force $\mathbf{F} = \mathbf{J} \times \mathbf{B}$ from the **Magnetic Fields** interface is passed to the **Solid Mechanics** interface and, passes the induced electric field $\mathbf{E} = \mathbf{v} \times \mathbf{B}$ from the **Solid Mechanics** interface to the **Magnetic Fields** interface. The Lorentz force effect on a moving structure is similar to that of the Rayleigh alpha-damping, where the damping coefficient is proportional to the background magnetic field and the material electric conductivity.

The Lorentz force and induced electric field can also be added separately from the interfaces, for instance, by using the **Body Load** feature from the **Solid Mechanics** interface and the **Velocity (Lorentz term)** feature (in 2D and 2D axisymmetric) from the **Magnetic Fields** interface. Once the **Magnetomechanics, Solid** feature is applied with this setting, those interface features with the same purpose should not be used to avoid double counting. When modeling electroacoustic transducers, like loudspeakers, the coupling is typically added in the voice coil domain.

The coupling feature handles frames as well as the correct formulation for transient and frequency domain problems. When the displacement of the vibrating or moving part of the model cannot be neglected (when topology changes are important), the **Moving Mesh** feature (added from the **Definitions**) is usually used together with the **Magnetomechanics, Solid** feature. In this case, the **Material Type** of the moving part in the **Magnetic Fields** interface has to set to **Solid**. In the **Moving Mesh** setting, the **Prescribed Mesh Displacement** is usually set as the dependent variables of the **Solid Mechanics** interface.

When the **Lorentz Coupling** is used in a domain with magnetic materials such as a magnetizable armature, the Lorentz force is no longer the only force contribution. In this case, the force contribution from the magnetic polarization might need to be considered. This can be done by adding a **Force Calculation** feature from the **Magnetic Fields** interface and adding a **Body Load** feature from the **Solid Mechanics** interface. Note that the **Force Calculation** feature includes the force contribution not only from magnetic polarization but also from the Lorentz force, that is, the total electromagnetic force. Therefore, when the coupling between the **Magnetic Fields** interface and **Solid Mechanics** interface is implemented with the **Force Calculation** feature and **Body Load** feature, the **Only use Lorentz force** setting should not be used in the same domain.

There are several library examples that demonstrate its usage:

	Loudspeaker Driver — Frequency-Domain Analysis: Application Library path **Acoustics_Module/Electroacoustic_Transducers/loudspeaker_driver**
	Loudspeaker Driver — Transient Analysis: Application Library path **Acoustics_Module/Electroacoustic_Transducers/loudspeaker_driver_transient**
	Magnetic Damping of Vibrating Conducting Solids: Application Library path **ACDC_Module/Electromagnetics_and_Mechanics/magnetic_damping**
	AC Contactor with Shading Coil: Application Library path **ACDC_Module/Electromagnetics_and_Mechanics/contactor_shading_coil**
	Deformation of an Iron Plate by Magnetic Force: Application Library path **ACDC_Module/Electromagnetics_and_Mechanics/plate_deflected_by_magnet**

Magnetomechanics, Boundary

Use the **Magnetomechanics, Boundary** ( ) coupling to model interaction between magnetic fields and deformable shells and membranes. It can be used to model cases with or without induction currents, depending on which magnetic field physics interface is used. Thus **Magnetic Fields** or **Magnetic Fields, No Currents** interfaces can be used and coupled with either structural **Shell** or **Membrane** interfaces. The coupling can also handle large structural deformations which can significantly affect the magnetic field distribution

The following study types are supported:

- Stationary
- Time Dependent
- Eigenfrequency
- Frequency-Domain Perturbation

The interaction is related to the action of the magnetics Maxwell stress which is a quantity that is quadratic in terms of magnetics fields. Therefore, to perform either eigenfrequency of frequency domain analysis, the presence of a stationary bias magnetic field will be needed. Thus, only Frequency-Domain Perturbation study type

is supported for frequency domain computations. You can use one of the following predefined studies which will include a stationary step to compute the bias state:

- Frequency Domain, Prestressed
- Eigenfrequency, Prestressed
- Small-Signal Analysis, Frequency Domain

The presence of this multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Magnetomechanics, Boundary 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `mmfb1`.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Thin structure** and **Magnetic Fields** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example

Membrane is deleted, then the **Thin structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual.*

BOUNDARY SELECTION

All applicable boundaries are selected automatically. Such domains represent an intersection of the applicable boundaries for the physics interfaces selected in the coupling feature.

In the **Shell** interface, the following boundary material features are applicable:

- **Linear Elastic Material**
- **Rigid Material**

In the **Membrane** interface, the following boundary material features are applicable:

- **Linear Elastic Material**
- **Nonlinear Magnetostrictive Material**
- **Nonlinear Elastic Material** (with the Nonlinear Structural Materials Module)
- **Hyperelastic Material** (with the Nonlinear Structural Materials Module)

In both the **Magnetic Fields** and **Magnetic Fields, No currents** interfaces, the following internal boundary conditions can be used for modeling thin layers:

- **Magnetic Shielding**. This boundary condition supports both linear and nonlinear (BH curve) magnetization models within the layer. Use this option for modeling of thin layers of ferromagnetic materials such as iron or steel. Note however that this boundary condition is only applicable for stationary study.
- **Thin Law Permeability Gap**. Use this option for modeling, for example, flexible thin layers of nonmagnetic material (such as plastic) separating domains of magnetic medium.

In addition, the **Magnetic Fields** interface provides on internal boundaries a **Transition Boundary Condition** that supports all study types. Use this option for modeling thin

layers of conductive material such as metals. The material can be also ferromagnetic. Note however that this boundary condition only supports linear magnetization model within the layer.

See The Magnetic Fields Interface and The Magnetic Fields, No Currents Interface in the *AC/DC Module User's Guide*.

LORENTZ COUPLING

This section is available only if one of the coupled interfaces is **Magnetic Fields**.

Select the **Only use Lorentz force** checkbox in order to model the interaction by only applying the Lorentz surface force $\mathbf{f} = \mathbf{J}_s \times \mathbf{B}$ as the load on the structure. This option is useful when modeling thin layers of electrically conductive but nonmagnetic material such as aluminum or copper. The Lorentz force effect on a moving structure is similar to that of the Rayleigh alpha-damping, where the damping coefficient is proportional to the background magnetic field and the material electric conductivity.

The interaction will be limited to the boundaries, where the coupled **Magnetic Fields** interface defines surface current $\mathbf{J}_s$. The following features are included:

- **Transition Boundary Condition**
- **Impedance Boundary Condition**
- **Surface Current Density**

Magnetic Forces, Rotating Machinery

Use **Magnetic Forces, Rotating Machinery** ( ) coupling to model interaction between deformable solids and rotating magnetic fields, particularly in case when the rotation and deformations of the structure can significantly affect the magnetic field distribution.

The presence of such multiphysics node in the model will enforce the **Include geometric nonlinearity** option on all applicable study nodes.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `mfrm1`.

DOMAIN SELECTION

The domain selection includes by default all applicable domains. Such domains represent an intersection of the applicable domains under the corresponding **Structure** and **Rotating Machinery, Magnetic** interfaces selected in the coupling feature.

In **Rotating Machinery, Magnetic** interface, the following domain features are applicable:

- **Ampère's Law**, if its material type input is set to **Solid**. Use this domain feature for solid magnetic materials.
- **Magnetic Flux Conservation**, if its material type input is set to **Solid**. Use this domain feature for solid magnetic materials.
- **Nonconducting Magnet**
- **Conducting Magnet**
- **Ampère's Law, Nonlinear Magnetostrictive**
- **Passive Conductor**
- **Coil**, if its material type input is set to **Solid**. You can use this feature to represent current carrying solid domains such as coils and wires.

In the **Structure** interface, the following domain material features are applicable:

- **Linear Elastic Material**
- **Nonlinear Magnetostrictive Material**
- **Nonlinear Elastic Material** (with the Nonlinear Structural Materials Module)

- **Hyperelastic Material** (with the Nonlinear Structural Materials Module)
- **Rigid Material**.

- For nonsolid magnetic domains, remove them from the **Structure** interface selection but keep them selected in the **Rotating Machinery, Magnetic** interface selection.
- **Ampère's Law, Nonlinear Magnetostrictive** is a part of The Nonlinear Magnetostriction Interface. The corresponding multiphysics coupling feature, Nonlinear Magnetostriction, can be combined with **Magnetic Forces, Rotating Machinery**.

See also

- Magnetomechanics

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Structure** and **Rotating Machinery, Magnetic** lists include all applicable physics interfaces.

Select **Structure** — **Solid Mechanics** or **Multibody Dynamics**.

Select **Rotating machinery, magnetic** — **Rotating Machinery, Magnetic**.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the coupling from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid**

Mechanics is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual.*

Piezomagnetism

The **Piezomagnetism** multiphysics node ( ) passes the appropriate magnetization contribution from the **Piezomagnetic Material** node in the **Solid Mechanics** interface (where it is defined together with the magnetoelastic properties of the material) to the **Ampère's Law, Piezomagnetic** node in the **Magnetic Fields** interface. It also passes the mechanics stress contribution due to the applied magnetic field back to the **Piezomagnetic Material** node.

Instead of **Magnetic Fields** interface, you can also use **Rotating Machinery, Magnetic** interface.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Piezomagnetism 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `pzm1`.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Solid mechanics** and **Magnetic fields** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when The Piezomagnetism Interface is selected in the **Model Wizard** or **Add Physics** window, then the participating Solid Mechanics and Magnetic Fields interfaces are selected.

You can also select **None** from either list to uncouple the Piezomagnetism node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

DOMAIN SELECTION

The domain selection includes by default all applicable domains. Such domains represent an intersection of the applicable domains under the corresponding **Magnetic Fields** and **Solid Mechanics** interfaces selected in the coupling feature.

In **Magnetic Fields** interface, the following domain feature is applicable:

- **Ampère's Law, Piezomagnetic**

In **Solid Mechanics** interface, the following domain feature is applicable:

- **Piezomagnetic Material**.

For nonsolid magnetic domains, remove them from the Solid Mechanics interface selection but keep them selected in the Magnetic Fields interface selection.

COUPLING TYPE

From the list, choose one of these coupling types:

- **Fully coupled** (the default) to include both the direct and inverse piezomagnetic effects.
- **Joule effect**, also known as direct piezomagnetic effect, to include only the deformation of the material caused by its magnetization in response to the applied magnetic field.

Villari effect, also known as inverse piezomagnetic effect, to include only the change in the material magnetization as a result of applied mechanical stress or strain.

Nonlinear Magnetostriction

The **Nonlinear Magnetostriction** multiphysics node ( ) passes the appropriate magnetization contribution from the **Linear Elastic Material** node in the **Solid Mechanics** interface to the **Ampère's Law, Nonlinear Magnetostrictive** node in the **Magnetic Fields** interface. It also passes the inelastic strain contribution due to the applied magnetic field back to the **Linear Elastic Material** node. The strains are quadratic in terms of the material magnetization.

Instead of **Magnetic Fields** interface, you can also use **Rotating Machinery, Magnetic** interface.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Nonlinear Magnetostriction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `pzm1`.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Solid mechanics** and **Magnetic fields** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when The Nonlinear Magnetostriction Interface is selected in the **Model Wizard** or **Add Physics** window, then the participating Solid Mechanics and Magnetic Fields interfaces are selected.

You can also select **None** from either list to uncouple the Nonlinear Magnetostriction node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

DOMAIN SELECTION

The domain selection includes by default all applicable domains. Such domains represent an intersection of the applicable domains under the corresponding **Magnetic Fields** and **Solid Mechanics** interfaces selected in the coupling feature.

In **Magnetic Fields** interface, the following domain feature is applicable:

- **Ampère's Law, Nonlinear Magnetostrictive**

In **Solid Mechanics** interface, the following domain material feature are applicable:

- **Linear Elastic Material**
- **Hyperelastic Material** (with the Nonlinear Structural Materials Module).

For nonsolid magnetic domains, remove them from the Solid Mechanics interface selection but keep them selected in the Magnetic Fields interface selection.

COUPLING TYPE

From the list, choose one of these coupling types:

- **Fully coupled** (the default) to include both the direct and inverse magnetostrictive effects.
- **Joule effect**, also known as direct magnetostrictive effect, to include only the deformation of the material caused by its magnetization in response to the applied magnetic field.

Villari effect, also known as inverse magnetostrictive effect, to include only the change in the material magnetization as a result of applied mechanical stress or strain.

MAGNETOSTRICTION

From the **Magnetostriction model** list, choose one of the following options:

- **Isotropic** (default)
- **Cubic crystal**

For the **Isotropic** choice, you can enter one dimensionless value of the **Saturation magnetostriction**, λ_s.

For the **Cubic crystal** choice, you can enter two dimensionless values of the **Magnetostriction constants**, λ_{100} and λ_{111} corresponding to different crystallographic planes.

In both cases, the default is to take the values **From material**. Also enter the magnitude of the **Saturation magnetization**, M_s.

When this coupling node is used as a part of The Nonlinear Magnetostriction Interface multiphysics interface, the **Ampère's Law, Nonlinear Magnetostrictive** feature under the corresponding **Magnetic Fields** interface also contains an input for the saturation magnetization. That input will then become inaccessible (grayed out), and the value entered at this coupling node will be used in all computations.

ADVANCED SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Option** in the **Show More Options** dialog.

You can select how the structural contribution due to the magnetostriction is accounted for. The two options are **Inelastic contribution to**: **Stress** (default) or **Strain**.

This setting is effective only when the coupling type is set to either Joule effect or fully coupled. In the case of the contribution to stress, you can select which deviatoric stress will contribute to the effective field. The options are: **Include elastic stress** (default) or **Include total stress**.

The second option was the default behavior for the magnetostriction in COMSOL versions prior to 6.1.

Magnetostriction

The **Magnetostriction** coupling feature is obsolete and cannot be added in version 6.1 and later. It may be present in models created by earlier versions of the software.

You are advised to update your model to use one of the new multiphysics coupling features instead, either Nonlinear Magnetostriction or Piezomagnetism (for modeling linear magnetostrictive effects).

Fluid–Structure Interaction

The **Fluid–Structure Interaction** ( ) multiphysics node provides a coupling on a boundary between a fluid domain and a solid material. The solid material can be modeled either in a neighboring domain, or on the boundary itself. In the former case, the Solid Mechanics or Multibody Dynamics interface is used; in the latter one of the Layered Shell, Shell, or Membrane interfaces is used.

The effect of the coupling depends on whether a **Deforming Domain** is active in the fluid domain or not. For cases when the structural deformations are so small that the change in the geometry of the fluid can be ignored, you do not have to use a deforming domain. This is called a *fixed geometry*. Domains with a fixed geometry have fewer degrees of freedom and are less nonlinear and are thus computationally less expensive

If **Deforming Domain** is active in any domain, the **Include geometric nonlinearity** option is automatically selected in all applicable study nodes.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Fluid–Structure Interaction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `fsi1`.

BOUNDARY SELECTION

By default, the selection is set to **All boundaries**. However, only boundaries between fluid and solid domains are applicable. This is deduced automatically using the selections made on the coupled physics interfaces.

COUPLED INTERFACES

This section defines the physics interfaces involved in the Fluid–Structure Interaction coupling. The **Fluid** and **Structure** lists include all applicable physics interfaces.

The default values depend on how the node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Fluid–Structure Interaction** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

COUPLING TYPE

In the case when the coupling is of the fixed geometry type, you can select which aspect of the interaction you want to model. From the **Fixed geometry coupling type** list, select **Fully coupled**, **Fluid loading on structure**, or **Velocity transmission to fluid**.

In the case **Fluid loading on structure**, pressure and viscous forces in the fluid affect the structure, but the structure does not affect the fluid flow.

In the case **Velocity transmission to fluid**, the structural velocity acts as boundary condition for the fluid, but the fluid does not affect the structure.

For **Fully coupled**, both effects are taken into account.

When a boundary is adjacent to an active deforming domain, the settings is this section are ignored.

See also:

- Fluid–Structure Interaction in the Structural Mechanics Modeling chapter
- Fluid–Structure Interaction

Fluid–Structure Interaction, Pair

The **Fluid–Structure Interaction, Pair** ( ) multiphysics node provides a coupling between a pair of boundaries, where one is the boundary of a fluid domain and the other is part of solid structure. The solid material can be modeled either in a neighboring domain or on the boundary itself. In the former case, the Solid Mechanics or Multibody Dynamics interface is used; in the latter, one of the Shell or Membrane interfaces is used.

You need to use this type of coupling if the geometry sequence is set up to form an assembly. In the more common case of a union when the boundary is shared, use the Fluid–Structure Interaction multiphysics coupling.

The effect of the coupling depends on whether a **Deforming Domain** is active in the fluid domain or not. For cases when the structural deformations are so small that the change in the geometry of the fluid can be ignored, you do not have to use a deforming

domain. This is called a *fixed geometry*. Domains with a fixed geometry have fewer degrees of freedom and are less nonlinear and are thus computationally less expensive

If **Deforming Domain** is active in any domain, the **Include geometric nonlinearity** option is automatically selected in all applicable study nodes.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Fluid–Structure Interaction, Pair 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `fsip1`.

BOUNDARY SELECTION

The boundary selection cannot be edited manually. The contents are obtained from the **Pair Selection**.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Fluid** and **Structure** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Fluid–Structure Interaction, Pair** node from a physics interface. If the physics interface is removed from the **Model**

Builder, for example **Solid Mechanics** is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

PAIR SELECTION

Select pairs for which this coupling is to be used. The pairs must be defined under **Definitions** in the component.

FIXED GEOMETRY

In the case when the coupling is of the fixed geometry type, you can select which aspect of the interaction you want to model. From the **Fixed geometry coupling type** list, select **Fully coupled**, **Fluid loading on structure**, or **Velocity transmission to fluid**.

In the case **Fluid loading on structure**, pressure and viscous forces in the fluid affect the structure, but the structure does not affect the fluid flow.

In the case **Velocity transmission to fluid**, the structural velocity acts as boundary condition for the fluid, but the fluid does not affect the structure.

For **Fully coupled**, both effects are taken into account.

When a boundary is adjacent to an active deforming domain, the settings is this section are ignored.

Fluid–Pipe Interaction

The **Fluid–Pipe Interaction** ( ) multiphysics node provides a way to transfer fluid loads from the Pipe Flow interface to the Pipe Mechanics interface.

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Fluid–Pipe Interaction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In

order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `fpipe1`.

EDGE SELECTION

By default, the selection is set to **All edges**. However, only edges that belong to both the Pipe Flow and Pipe Mechanics interfaces are applicable. This is deduced automatically using the selections made on the coupled physics interfaces.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Fluid** and **Structure** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Fluid–Pipe Interaction** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Pipe Mechanics** is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual.*

FLUID LOAD

In this section, you specify the load contributions that are transferred from the Pipe Flow interface to the Pipe Mechanics interface.

Select the **Drag Force** checkbox to include the forces from viscous drag along the pipe.

Select the **Centrifugal Force** checkbox to include centrifugal forces from the fluid in curved pipes.

Select the **Pipe flow pressure** checkbox to include effects of the internal pressure in the pipe. When **Pipe flow pressure** is selected, you can also enter an external pressure. Select a method for entering the **External pressure**, p_0 — **User defined** or **Common model input**.

For **User defined**, enter a value or an expression for the external pressure.

Select the **Junction forces** checkbox to include local forces occurring at various connections and components in the piping system.

The checkboxes in this section interact with any Fluid Load node on the same selection. When a checkbox is selected, the corresponding entry in the **Fluid Load** node is made unavailable.

See also:

- Effects of Internal Pressure and Fluid Flow Forces in the Theory for the Pipe Mechanics Interface section.
- Fluid Load in The Pipe Mechanics Interface section.

Structure–Thin-Film Flow Interaction

The **Structure–Thin-Film Flow Interaction** ( ) multiphysics node provides a coupling on a boundary between a thin fluid film and a solid material.

The **Structure–Thin-Film Flow Interaction** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Structure–Thin-Film Flow Interaction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `stfi1`.

BOUNDARY SELECTION

By default, the selection is set to **All boundaries**. However, only boundaries between fluid and solid domains are applicable. This is deduced automatically using the selections made on the coupled physics interfaces.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Structure** and the **Thin film** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Structure–Thin-Film Flow Interaction** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

Shell–Thin-Film Flow Interaction

The **Shell–Thin-Film Flow Interaction** ( ) multiphysics node provides a coupling on a boundary between a thin fluid film and a shell.

The **Shell–Thin-Film Flow Interaction** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `Shell–Thin-Film Flow Interaction 1`.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `shtfi1`.

BOUNDARY SELECTION

By default, the selection is set to **All boundaries**. However, only boundaries between fluid and solid domains are applicable. This is deduced automatically using the selections made on the coupled physics interfaces.

COUPLED INTERFACES

This section defines the physics interfaces involved in this multiphysics coupling. The **Structure** and the **Thin film** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Shell–Thin-Film Flow Interaction** node from a physics interface. If the physics interface is removed from the **Model**

Builder, for example **Shell** is deleted, then the **Structure** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

FILM LOCATION

In this section, you specify the load contributions that are transferred from the Thin-Film Flow interface to the Shell interface.

Select **Top** to include the load from the fluid on the top side of the shell.

Select **Bottom** to include the load from the fluid on the bottom side of the shell.

Phase-Field Damage (Multiphysics Coupling)

The **Phase-Field Damage** multiphysics coupling node ( ) links the Solid Mechanics interface bidirectionally with the Phase Field in Solids interface to model the evolution of damage and cracks in deforming solids.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `pfdmg1`.

DOMAIN SELECTION

When the node is added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains where the phase–field damage coupling is defined, or select **All domains** as needed.

When the **Phase-Field Damage** multiphysics interface is added from the **Model Wizard** or the **Add Physics** window, the domain selection of the coupling node is set to **All domains** by default.

Only domains that are active in the physics features selected in the Material Models section can be selected. The selection is only applicable when the participating Phase Field in Solids interface contains a single phase field variable.

COUPLED INTERFACES

This section defines the physics involved in this multiphysics coupling. The **Solid mechanics** and **Phase field in solids** lists include all applicable physics interfaces.

The default values depend on how the coupling node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (macOS and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a multiphysics interface is selected in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **Phase-Field Damage** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example if **Phase Field in Solids** is deleted, then the **Phase field in solids** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to reestablish the coupling, you need to choose the physics interface again from the **Solid mechanics** or **Phase field in solids** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Workflow in the *COMSOL Multiphysics Reference Manual*.

MATERIAL MODELS

In the **Parent material model** list, choose the material model for which damage is applied. All **Linear Elastic Material** and **Hyperelastic Material** subnodes of the coupled Solid Mechanics interface are listed here. If the fallback option **None** is selected, the domain selection is not applicable.

Similarly, in the **Parent phase field model** list, select the **Phase Field Model** subnode of the coupled Phase Field in Solids interface that controls the phase field evolution equation. If the fallback option **None** is selected, the domain selection is not applicable.

CRACK DRIVING FORCE

Select the **Crack driving force** — **Elastic strain energy density**, **Total strain energy density**, **Rankine criterion**, **Principal stress criterion**, or **User defined**. See Crack Driving Force for details.

- For **Elastic strain energy density** and **Total strain energy density**, enter the **Critical energy release rate** G_c. The default for the **Critical energy release rate** is to take its value **From material**. Choose **User defined** to enter another value or expression in the text field. If the **Parent phase field model** is of the AT2 type, it is also possible to enter a **Strain energy threshold** G_{0c}.
- For the **Rankine criterion**, enter the **Critical energy release rate** G_c.
- For the **Principal stress criterion**, enter the **Critical fracture stress** σ_{cr} and the post-peak slope parameter ξ. Choose **User defined** to enter another value or expression in the text field. This criterion is only available when the **Parent phase field model** is of the AT2 type.
- For **User defined**, enter an expression for the crack driving force D_d (dimensionless).

In the **Initial crack driving force** list, select **Automatic** or **Manual**. The **Automatic** option computes the initial value for the dimensionless crack driving force as $H_{d,0} = Q'(0)/d'(0)$, which is required for admitting a homogeneous zero solution for the phase field based on the phase field potential $Q(\phi)$ and the damage evolution function $d(\phi)$ chosen. For example, for the AT2 phase field model the initial value is 0, while it becomes $3/16$ for the AT1 model with a **Quadratic** damage evolution function. Choose **Manual** to enter another value or expression for the initial value (default 0).

In the **Exclude compressive energy** list, select a method for controlling which part of the strain energy density contributes to the crack driving force. This method also controls how the damaged stress tensor is constructed. See Strain Energy Split for details.

- If the **Parent material model** is a **Linear Elastic Material**, choose between **Volumetric only**; **Spectral decomposition, stress**; **Spectral decomposition, strain**; or **No split**. The option **Spectral decomposition, strain** is only available if the linear elastic material is isotropic.
- If the **Parent material model** is a **Hyperelastic Material**, choose between **Volumetric only** or **No split**. The option **Volumetric only** is only available if the hyperelastic

material permits a clear split between the volumetric and isochoric strain energy density.

DAMAGE EVOLUTION

Select the **Damage evolution** function $d(\phi)$ — **Quadratic**; **Cubic, Borden**; **Rational**; **Linear softening**; **Exponential softening**; **Cornelissen softening**; or **User defined**. See Damage Evolution for Phase-Field Damage for details.

- For **Quadratic**, the damage evolution function (dimensionless) is defined as $d(\phi) = 1 - (1-\phi)^2$. If this option is used together with the **PF–CZM** phase field model, a constraint on the phase field might be required to ensure a unique solution.
- For **Cubic, Borden**, enter a value for the **Model parameter** s. This option is only available if the **Crack driving force** is **Elastic strain energy density** or **Total strain energy density** and the **Parent phase field model** is of the AT2 type.
- For **Rational**, enter values for the model parameters p, a_1, a_2, and a_3. The damage evolution function is then defined as $d(\phi) = Q(\phi)/((1-\phi)^p + Q(\phi))$, with $Q(\phi) = a_1\phi(1 + a_2\phi(1 + a_3\phi))$. The value for the exponent p should be greater than or equal to 2.
- The options **Linear softening**, **Exponential softening**, and **Cornelissen softening** are special cases of the **Rational** damage evolution function intended for modeling quasi-brittle fracture inspired by cohesive zone models. The parameter a_1 is computed from the **Critical stress** σ_{cr}, while p, a_2, and a_3 are predetermined constants.
- For **User defined**, enter an expression for the damage evolution $d(\phi)$ as a function of the phase field variable (dimensionless).

Enter the **Maximum damage**, d_{max}, which controls the residual stiffness of the material when fully damaged. The default value for numerical stability is `1-1e-5`.

See also

- Modeling Damage in the *Structural Mechanics Modeling* chapter.
- Theory for the Phase Field in Solids Interface and Phase-Field Damage Models in the *Structural Mechanics Theory* chapter.

	The Multiphysics interface **Phase-Field Damage**, is only available with some COMSOL products. For a detailed overview of the functionality available in each product, visit www.comsol.com/products/specifications/

	Phase-Field Modeling of Dynamic Crack Branching: Application Library path **Nonlinear_Structural_Materials_Module/Damage/dynamic_crack_branching**

Couplings Between Structural Mechanics Interfaces

Some of the structural mechanics interfaces are formulated using different types of degrees of freedom. This needs to be considered in models where such interfaces interact. You can set up common transitions using the following predefined multiphysics couplings:

- Solid–Thin Structure Connection
- Solid–Beam Connection
- Shell–Beam Connection
- Layered Shell–Structure Transition
- Layered Shell–Structure Cladding
- Structure–Pipe Connection
- Embedded Reinforcement
- Beam Cross Section–Beam Coupling
- Beam–Beam Cross Section Coupling

Solid–Thin Structure Connection

Add the **Solid–Thin Structure Connection** multiphysics node ( ) to create transitions between domains modeled using the Solid Mechanics or Multibody Dynamics interfaces and boundaries modeled using the Shell or Membrane interfaces.

The **Solid–Thin Structure Connection** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

COUPLED INTERFACES

Select **Solid mechanics** — **Solid Mechanics** or **Multibody Dynamics**.

Select **Thin structure** — **Shell** or **Membrane**.

CONNECTION SETTINGS

When **Membrane** is selected from the **Thin structure** list, the **Connection type** is set to **Shared boundaries**. See settings in Shared Boundaries.

When **Shell** is selected from the **Thin structure** list, select the **Connection type** — **Solid boundaries to shell edges**, **Shared boundaries**, or **Parallel boundaries**.

For the first two options, there is an automatic search for possible adjacent geometrical objects. If you want to modify the selections, select the **Manual control of selections** checkbox. If you clear it, the selections will be replaced by the automatic ones.

Solid Boundaries to Shell Edges

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding shell edges in the **Edge Selection, Shell** section.

Select **Connected area defined by** — **Shell thickness**, **Selected solid boundaries**, or **Distance from shell midsurface**. This parameter determines how much of the selected solid boundaries that are connected to the shell. The default is that a distance from the shell edge having the size of the half the shell thickness in both perpendicular directions is connected.

- Using **Selected solid boundaries** connects the selected boundaries in their entirety to the shell.
- If you select **Distance from shell midsurface**, enter a **Distance** d. This value is used instead of the half the shell thickness for defining the connection distance.

Select a **Method** — **Rigid** or **Flexible**.

- The **Rigid** version of the coupling only adds constraints to the boundary of the solid, which in general causes local disturbances of the stress field, since the thickness cannot change.
- When using the **Flexible** coupling, three extra degrees of freedom are added along the shell edge. This allows for a more accurate description of the transition, but the model can in some cases become unconstrained if the mesh on the solid is very coarse.

When either of the **Shell thickness** or **Distance from shell midsurface** options are used, the connected area is slightly extended by the distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the shell thickness, which allows for small inaccuracies on for example a curved geometry. By selecting **User defined**, you can modify the value of Δ.

Shared Boundaries

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected boundaries in the **Boundary Selection** section.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the shell thickness, which allows for small inaccuracies on for example a curved geometry. By selecting **User defined**, you can modify the value of Δ.

Parallel Boundaries

Select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding shell boundaries in the **Boundary Selection, Shell** section.

Select **Distance evaluation** — **Shell properties**, **Solid–shell geometrical distance**, or **User defined**. This parameter controls how the coupling determines the connection distance, used when expressing the coupling between translation and rotation.

- When **Shell properties** is selected, the distance is based on the shell thickness and offset, so that a distance equal to half the shell thickness is used.
- If you select **Solid–shell geometrical distance**, the connection distance is computed from the geometrical distance between the selected boundaries.
- For **User defined**, enter a **Distance** d, which defines the connection distance.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the shell thickness, which allows for small inaccuracies on for example a curved geometry. By selecting **User defined**, you can modify the value of Δ.

- For more information about coupling different element types, see Coupling Techniques.
- For details about the formulation of this coupling, see Connection Between Shells and Solids

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select where to place the **Constraints on** — **Thin structure boundaries** or **Solid boundaries**.

By default the constraints are placed on the thin structure boundaries (Shell or Membrane), but it is also possible to enforce the constraints on the solid boundaries. This option is not available when Solid Boundaries to Shell Edges is selected from the **Connection type** list.

When Solid Boundaries to Shell Edges or Parallel Boundaries is selected from the **Connection type** list, an additional **Search method** option is available for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust mesh search method. The **Manual** method is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

These types of couplings will commonly generate many constraints, possibly even conflicting with other constraints coming from ordinary boundary conditions. For resolving potential problems, see

- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions

When there are many interconnected constraints, it may also be useful to change Null-Space Function to **Orthonormal** or **Explicit — Orthonormal**, but doing so may increase the memory consumption significantly.

An example of couplings between shells and solids is shown in *Connecting Shells and Solids*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/shell_solid_connection**

Solid–Beam Connection

Add the **Solid–Beam Connection** multiphysics coupling node ( ) to create transitions between domains modeled using the Solid Mechanics or Multibody Dynamics interfaces and edges modeled using the Beam interface.

The 2D and 3D versions of the **Solid–Beam Connection** are fundamentally different, and are treated separately below.

COUPLED INTERFACES

Select **Solid mechanics** — **Solid Mechanics** or **Multibody Dynamics**.

Select the **Beam** interface to couple.

CONNECTION SETTINGS 2D

Select the **Connection type** — **Solid boundaries to beam points**, **Solid and beam shared boundaries**, **Solid and beam parallel boundaries**, or **Solid edges to beam points**.

In first two cases, there is an automatic search for possible adjacent geometrical objects. If you want to modify the selections, select the **Manual control of selections** checkbox. If you clear it again, the selections will be replaced by the automatic ones.

Solid Boundaries to Beam Points

For a manual selection, select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding beam points in the **Point Selection, Beam** section.

Select **Connected area defined by** — **Section height**, **Selected boundaries**, or **Distance from beam axis**. This parameter determines how much of the selected solid boundaries that are actually connected to the beam. The default is that a distance from the beam point having the size of the half the beam section height in each direction is connected. Using **Selected boundaries** connects the entire selected boundaries to the beam. If you select **Distance from beam axis**, enter a **Distance** d. This is used instead of the section height for defining the connection distance.

Select a **Method** — **Rigid** or **Flexible**. The **Rigid** version of the coupling only adds constraints to the boundary of the solid, which in general causes local disturbances of the stress field, since the thickness cannot change. When using the **Flexible** coupling, three extra degrees of freedom are added to each beam point. This allows for a more accurate description of the transition, but the model can in some cases become unconstrained if the mesh on the solid is very coarse.

When the connected area is determined by either the **Section height** or **Distance from beam axis options**, the connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the beam height, which allows for small inaccuracies on for example a curved geometry. By selecting **User defined**, you can modify the value of Δ.

Solid and Beam Shared Boundaries

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected boundaries in the **Boundary Selection** section.

The connected area is slightly extended by the distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. The **Automatic** option sets the value of the tolerance Δ to 0.5% of section height of the beam. By selecting the **User defined** option, you can modify the value of the tolerance Δ.

Solid and Beam Parallel Boundaries

Select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding beam edges in the **Boundary Selection, Beam** section.

Select **Distance evaluation** — **Section height**, **Geometrical distance**, or **User defined**. This parameter determines how the coupling treats the determination of the connection distance, used when expressing the coupling between translation and rotation. The default is that a distance equal to half the section height is used. If you select **Geometrical distance**, the connection distance is computed from the geometrical distance between the solid boundary and the beam boundary. For **User defined** enter a **Distance** d. This defines the connection distance.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the beam height, which allows for small inaccuracies on, for example, a curved geometry. By selecting **User defined**, you can modify the value of Δ.

CONNECTION SETTINGS 3D

Select the **Connection type** — **Solid boundaries to beam points, general**, **Solid boundaries to beam points, transition**, **Solid boundaries to beam edges**, or **Solid edges to beam points**. For the connections to beam points, there is an automatic search for possible adjacent geometrical objects.

Solid Boundaries to Beam Points, General

Select this when the end of a beam is to be "welded" to part of the face of a solid domain.

If you want to modify the selections, select the **Manual control of selections** checkbox. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and select the corresponding beam points in the **Point Selection, Beam** section. If you clear the checkbox, the selections will be replaced by the automatic ones.

Select a **Connected region** — **Selected boundaries**, **Distance (automatic)**, **Distance (manual)**, or **Connection criterion**.

- Using **Selected boundaries** makes all selected boundaries to be rigidly connected to the point on the beam.
- For **Distance (automatic)**, all parts on the solid boundary, which are within the default distance from the beam point, are connected. This distance is determined by the cross section properties of the beam. It is contained in the variable `beam.re` as described in the documentation for The Beam Interface.
- For **Distance (manual)** enter a **Connection radius** r_c. All parts on the solid boundary, which are within the given distance from the beam point, are connected.
- For **Connection criterion** enter a Boolean expression in the text field. The beam is connected to the selected solid boundaries, wherever the expression has a nonzero value. The default value is 1, which is equivalent to using the **Selected boundaries** option.

Solid Boundaries to Beam Points, Transition

Use this option if you want to model a transition from a beam modeled by solid elements to the same beam modeled using beam elements. It is assumed that the solid part is truncated by a set of boundaries which are perpendicular to the beam axes. No settings other than the selections are needed in this case.

If you want to modify the selections, select the **Manual control of selections** checkbox. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and select the corresponding beam points in the **Point Selection, Beam** section. If you clear the checkbox, the selections will be replaced by the automatic ones.

The coupling can introduce a set of extra degrees of freedom on the solid boundary for computing the warping displacement over the cross section. Select **Include warping** — **Automatic**, **Yes**, or **No**, to control this behavior. If **Automatic** is selected, warping will be included, unless the beam cross section is one of the predefined sections **Circular** or **Pipe**. To explicitly control the behavior, select **Yes** or **No**.

The warping degrees of freedom should preferably be solved for only once, and separately. If they are solved at the same time as the general problem, you can expect that the solution time increases by a factor of two. For more details, see Beam Point to Solid (3D) in the Structural Mechanics Modeling chapter.

Solid Boundaries to Beam Edges

Use this option when you want to model a beam that is parallel to the solid boundary and is "welded" to it.

Select a **Connected region** — **Selected boundaries**, **Beam width**, or **Connection criterion**.

- Using **Selected boundaries** makes all selected boundaries to be rigidly connected to the point on the beam. This is mainly useful if the boundary has approximately the same shape as the physical size of the beam, such as a rectangle with the same width as a beam flange.
- For **Beam width**, enter a **Width** w. All parts on the solid boundary, which are within the given distance from the projection of the beam axis on the solid, are connected. For cases when the beam is asymmetric, that is when the connection distance from the beam axis is not the same in both directions, enter also an **Offset** d. The positive orientation of the offset is $\mathbf{n} \times \mathbf{t}$, where $\mathbf{n}$ is the outward normal of the solid, and $\mathbf{t}$ is the positive orientation of the beam edge.
- For **Connection criterion** enter a Boolean expression in the text field. The beam is connected to the selected solid boundary, wherever the expression has a nonzero value. The default value is 1, which is equivalent to using the **Selected boundaries** option.

Solid Edges to Beam Points

Select this when the end of a beam is to be "welded" to an edge of a solid domain. One important use case is when modeling bolts using beams, and this coupling acts as an abstract bolt head.

Select the connected solid edges in the **Edge Selection, Solid** section and select the corresponding beam points in the **Point Selection, Beam** section.

Select a **Connected region** — **Selected edges**, **Distance (automatic)**, **Distance (manual)**, or **Connection criterion**.

- Using **Selected edges** makes all selected edges to be rigidly connected to the point on the beam.
- For **Distance (automatic)**, all parts on the solid edges, which are within the default distance from the beam point, are connected. The distance is determined by the

cross section properties of the beam. It is contained in the variable `beam.re` as described in the documentation for The Beam Interface.

- For **Distance (manual)** enter a **Connection radius** r_c. All parts on the solid edges, which are within the given distance from the beam point, are connected.
- For **Connection criterion** enter a Boolean expression in the text field. The beam is connected to the selected solid edges, wherever the expression has a nonzero value. The default value is 1, which is equivalent to using the **Selected edges** option.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The **Search method** shows available mesh search methods for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust search method. The **Manual** method is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

- For more information about coupling different element types, see Coupling Techniques.
- For details about the formulation of this coupling, see Connection Between Shells and Solids

These types of couplings will commonly generate many constraints, possibly even conflicting with other constraints coming from ordinary boundary conditions. For methods of resolving potential problems, see

- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions

When there are many interconnected constraints, it may also be useful to change Null-Space Function to **Orthonormal** or **Explicit — Orthonormal**, but doing so may increase the memory consumption significantly.

Shell–Beam Connection

Add the **Shell–Beam Connection** multiphysics node ( ) to create transitions between boundaries or edges modeled using the Shell interface and edges or points modeled using the Beam interface.

The **Shell–Beam Connection** node is only available with some COMSOL products (see www.comsol.com/products/specifications/).

CONNECTION SETTINGS

Select the **Connection type** — **Shell edges to beam points**, **Shared edges**, **Parallel edges**, or **Shell boundaries to beam points**.

For all options except **Parallel edges**, there is an automatic search for possible adjacent geometrical objects. If you want to modify the selections, select the **Manual control of selections** checkbox. If you clear it again, the selections will be replaced by the automatic ones.

Shell Edges to Beam Points

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected shell edges in the **Edge Selection, Shell** section and the corresponding beam points in the **Point Selection, Beam** section.

Select a **Connected region** — **Selected edges**, **Distance (automatic)**, **Distance (manual)**, or **Connection criterion**.

- Using **Selected edges** makes all selected edges rigidly connected to the point on the beam.
- For **Distance (automatic)**, all parts on the shell edge, which are within the default distance from the beam point, are connected. This distance is determined by the cross section properties of the beam. It is contained in the variable `beam.re` as described in the documentation for The Beam Interface.
- For **Distance (manual)** enter a **Connection radius** r_c. All parts on the shell edge, which are within the given distance from the beam point, are connected.
- For **Connection criterion** enter a Boolean expression in the text field. The beam is connected to the selected shell edge wherever the expression has a nonzero value. The default value is 1, which is equivalent to using the **Selected edges** option.

Shared Edges

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected edges in the **Edge Selection** section.

Select an **Offset definition** — **Along shell normal** or **Offset vector**.

- For **Along shell normal** enter an **Offset** ζ.
- For **Offset vector** enter values for $\mathbf{d}_0$ in the table. The offset is the vector from the reference surface of the shell to the actual position of the beam. From which the projection onto the shell normal will be extracted.

Parallel Edges

Enter a value for the **Parallelism tolerance** $\Delta\theta$. The default is 1 degree. The two edges are connected only where they are parallel within the specified tolerance.

Shell Boundaries to Beam Points

If the automatic selection is not sufficient, select the **Manual control of selections** checkbox. Then, select the connected shell boundaries in the **Boundary Selection, Shell** section and the corresponding beam points in the **Point Selection, Beam** section.

Select an option from the **Connected region** list — **Selected boundaries**, **Distance (automatic)**, **Distance (manual)**, or **Connection criterion**.

- Using **Selected boundaries** makes all selected boundaries rigidly connected to the point on the beam.
- For **Distance (automatic)**, all parts on the shell boundary, which are within a default distance from the beam point, are connected. This distance is determined by the cross section properties of the beam. It is contained in the variable `beam.re` as described in the documentation for The Beam Interface.
- For **Distance (manual)** enter a **Connection radius** r_c. All parts on the shell edge, which are within the given distance from the beam point, are connected.
- For **Connection criterion** enter a Boolean expression in the text field. The beam is connected to the selected shell boundary wherever the expression has a nonzero value. The default value is 1, which is equivalent to using the **Selected boundaries** option.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The **Search method** shows available mesh search methods for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust search method. The **Manual** method is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

- For more information about coupling different element types, see Coupling Techniques.
- For details about the formulation of this coupling, see Connection Between Shells and Solids

These types of couplings will commonly generate many constraints, possibly even conflicting with other constraints coming from ordinary boundary conditions. For methods of resolving potential problems, see

- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions

When there are many interconnected constraints, it may also be useful to change Null-Space Function to **Orthonormal** or **Explicit — Orthonormal**, but doing so may increase the memory consumption significantly.

Examples of all types of couplings between shells and beams are shown in *Connecting Shells and Beams*: Application Library path **Structural_Mechanics_Module/Beams_and_Shells/shell_beam_connection**

Layered Shell–Structure Transition

Add the **Layered Shell–Structure Transition** multiphysics node ( ) to create transitions between boundaries modeled using the Layered Shell and domains or boundaries modeled using the Solid Mechanics or Shell interfaces.

The **Layered Shell–Structure Transition** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

COUPLED INTERFACES

Select the **Layered Shell** interface to couple.

Select the **Structure** — **Solid Mechanics** or **Shell**.

SHELL PROPERTIES

Clear the **Use all layers** checkbox to apply the **Layered Shell–Structure Transition** coupling on some layers only. Select a **Layer** from the list (the coupling is then applicable only on the boundaries where this latter material is defined). Under the **Selection** table clear the checkboxes corresponding to layers where the coupling should not be applied.

CONNECTION SETTINGS

There is an automatic search for possible adjacent geometrical objects. If you want to modify the selections, select the **Manual control of selections** checkbox. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding shell edges in the **Edge Selection** section.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the shell thickness, which allows for small inaccuracies on, for example, a curved geometry. By selecting **User defined**, you can modify the value of Δ.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The **Search method** shows available mesh search methods for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust search method. The **Manual** method is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

Layered Shell–Structure Cladding

Add the **Layered Shell–Structure Cladding** multiphysics node ( ) to create claddings between boundaries modeled using the Layered Shell and domains or boundaries modeled using the Solid Mechanics, Membrane, or Shell interfaces.

The **Layered Shell–Structure Cladding** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

COUPLED INTERFACES

Select the **Layered Shell** interface to couple.

Select the **Structure** — **Solid Mechanics**, **Shell**, or **Membrane**.

CONNECTION SETTINGS

Select the **Connection type** — **Shared boundaries** or **Parallel boundaries**.

Shared Boundaries

There is an automatic search for possible adjacent geometrical objects. If you want to modify the selections, select the **Manual control of selections** checkbox. Select the boundaries where the cladding is applied in the **Boundary Selection** section. If you clear it, the selections will be replaced by the automatic ones.

Select the **Layered shell boundary** — **Top** or **Bottom** to apply the cladding.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set to 0.5% of the shell thickness, which allows for small inaccuracies on for example a curved geometry. By selecting **User defined**, you can modify the value of Δ.

Parallel Boundaries

Select the connected solid boundaries in the **Boundary Selection, Layered Shell** section and the corresponding structural boundaries in the **Boundary Selection, Structure** section.

Select the **Layered shell boundary** — **Top** or **Bottom** to apply the cladding.

In case of a shell structure, select the **Shell boundary** — **Top** or **Bottom** to apply the cladding.

The connected area is slightly extended by a distance Δ. Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance Δ is set

to 0.5% of the shell thickness, which allows for small inaccuracies on, for example, a curved geometry. By selecting **User defined**, you can modify the value of Δ.

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

Select where to place the **Constraints on** — **Layered shell boundaries** or **Structure boundaries**.

By default the constraints are placed on the Layered Shell side of the connection, but it is also possible to enforce the constraints on the other side (Solid Mechanics, Shell, or Membrane).

When Parallel Boundaries is selected from the **Connection type** list, the **Search method** option shows available mesh search methods for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust search method for the mapping operator. The **Manual** method is an alternative search method which is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

These types of couplings will commonly generate many constraints, possibly even conflicting with other constraints coming from ordinary boundary conditions. For methods of resolving potential problems, see

- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions

When there are many interconnected constraints, it may also be useful to change Null-Space Function to **Orthonormal** or **Explicit — Orthonormal**, but doing so may increase the memory consumption significantly.

An example of couplings between layered shells and solids is shown in *Connecting Layered Shells with Solids and Shells*: Application Library path **Composite_Materials_Module/Tutorials/layered_shell_structure_connection**

Layered Shell–Shell Connection

Add the **Layered Shell–Shell Connection** multiphysics node ( ) to create models where a combination of the Layered Shell interface and the Shell interface are used for different layers in the same layup. Thus, both interfaces are active on the same boundary. This type of modeling (*multiple model method*) is efficient in components, where some layers are thin, while other layers are significantly thicker.

The **Layered Shell–Shell Connection** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

COUPLED INTERFACES

Select the **Layered Shell** interface and **Shell** interfaces to couple.

INTERFACE SELECTION

Select **Apply to** — **Interior interfaces** or **Selected interfaces**.

For **Interior interfaces**, any interface shared between the two type of element formulations will be bonded.

For **Selected interfaces**, you have the possibility to select the interfaces to be bonded. In this case, select the applicable interfaces from the **Layer** list.

Analysis of a Composite Blade Using a Multiple Model Method: Application Library path **Composite_Materials_Module/Tutorials/composite_blade_multiple_model_method**

Structure–Pipe Connection

Add the **Structure–Pipe Connection** multiphysics coupling node ( ) to create transitions between edges modeled using the Pipe Mechanics interface and domains or boundaries modeled using the Solid Mechanics or Shell interfaces in 3D.

The multiphysics coupling is intended for situations where the cross-sectional properties match on both sides of the connection. It can be considered as an extension to the transition type couplings in Solid–Beam Connection and Shell–Beam Connection that also accounts for radial deformation of the pipe caused by the fluid pressure and the temperature difference over the cross section.

COUPLED INTERFACES

Select **Structure** — **Solid Mechanics** or **Shell**.

Select the **Pipe Mechanics** interface to couple.

- The connection assumes that the pipe cross section is circular. If another cross section is used, it is converted to an equivalent circular cross section
- For user defined pipe cross sections, the effects of fluid pressure and temperature are not considered
- Warping is not considered

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Options** in the **Show More Options** dialog.

The **Search method** shows available mesh search methods for the mapping operator between the source and destination boundaries.

The **Closest point** method is the default and most robust search method. The **Manual** method is computationally efficient but its robustness is determined by the **Search distance** and the **Extrapolation tolerance**.

See Advanced Settings for Nonlocal Couplings in the *COMSOL Multiphysics Reference Manual* for more information.

For more information about coupling different element types, see Coupling Techniques.

These types of couplings will commonly generate many constraints, possibly even conflicting with other constraints coming from ordinary boundary conditions. For methods of resolving potential problems, see

- Elemental and Nodal Constraints
- Suppressing Constraints on Lower Dimensions

When there are many interconnected constraints, it may also be useful to change Null-Space Function to **Orthonormal** or **Explicit — Orthonormal**, but doing so may increase the memory consumption significantly.

Embedded Reinforcement

Add an **Embedded Reinforcement** multiphysics node ( ) to create a coupling where elements from the Truss, Wire, Beam, or Membrane interface are embedded in solid elements. **Embedded Reinforcement** can also be used to couple the Truss, Explicit Dynamics and Solid Mechanics, Explicit Dynamics interfaces, but in that case, only these two interfaces can be connected.

The meshes used in the coupled interfaces do not have to coincide; the displacements inside the solid elements at the locations where the nodes of the lower-dimension element are located are automatically identified.

See Embedded Elements in the Structural Mechanics Theory Chapter.

The **Embedded Reinforcement** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

COUPLED INTERFACES

Standard Interfaces

Select the **Solid** interface to couple — **Solid Mechanics** or **Multibody Dynamics**.

Select the **Embedded structure** to couple — **Truss**, **Wire**, **Beam**, or **Membrane**.

When coupling the Beam interface to a Solid Mechanics or Multibody Dynamics interface, only the displacement degrees of freedom of the beam are constrained.

Explicit Dynamics Interfaces

Select the **Solid** interface to couple — **Solid Mechanics, Explicit Dynamics**.

Select the **Embedded structure** to couple — **Truss, Explicit Dynamics**.

CONNECTION SETTINGS

Select the **Connection type** — **Rigid**, **Spring constant per reference length**, or **Spring constant per reference area**. The available types of spring connections depend on the selected **Embedded Structure** interface.

When the **Embedded structure** is a Beam interface, select whether to **Suppress rotation around beam axis** or not. The default is to suppress axial rotation, to avoid rigid body rotation around the beam axis.

Rigid Connection

The **Rigid** connection type couples the selected interfaces using pointwise constraints applied to the selection of the embedded structure. The constraints are only active for the parts of the selection that lies within the solid.

Spring connection

All spring connection types couple the selected interfaces by inserting springs between the **Embedded structure** and the **Solid**. The properties of the spring connection are determined by user defined spring constants.

When the **Embedded structure** is a Truss or Wire interface, enter the **Axial spring constant** k_a, and the **Transverse spring constant** k_t. The default unit and expression for the spring constants depend on the **Connection type**:

- For **Spring constant per reference length**, the default expression for both k_a and k_t is `1e3*<tag>.Eequ*<tag>.area/h^2`.
- For **Spring constant per reference area**, the default expression for both k_a and k_t is `1e3*<tag>.Eequ*<tag>.perimeter/h^2`.

The variable `<tag>.Eequ` is a placeholder for the equivalent stiffness of the Truss or Wire interface, and `<tag>.area` and `<tag>.perimeter` for the cross-sectional area and perimeter, respectively. The multiplier `1e3` can be modified to tune the stiffness of the connection. Both connection types internally use the same formulation, and the spring constant per reference area is converted to a spring constant per reference length.

When the **Embedded structure** is a Beam interface and **Spring constant per reference length** is selected, enter the **Axial spring constant** k_a, and two **Transverse spring constants** in the local coordinate system of the beam, k_{yl} and k_{zl}. The default expression for the spring constants is `1e3*<tag>.Eequ*<tag>.area/h^2`.

The variable `<tag>.Eequ` is a placeholder for the equivalent stiffness of the Beam interface, and `<tag>.area` for the cross-sectional area. The multiplier `1e3` can be modified to tune the stiffness of the connection.

When the **Embedded structure** is a Membrane interface and **Spring constant per reference area** is selected, enter the three components of the stiffness vector in the boundary system coordinates, k_{t1}, k_{t2}, and k_n. The default expression for each component is `1e5*<tag>.Eequ*<tag>.d/h^2`.

The variable `<tag>.Eequ` is a placeholder for the equivalent stiffness of the Membrane interface, and `<tag>.d` for the thickness. The multiplier `1e5` can be modified to tune the stiffness of the connection.

Bond Slip Model

When a spring connection type is selected, it is also possible to also include a **Bond slip model**. By default, a **No** bond slip model is added. If **Friction** is selected, it is also possible to model sliding between the **Solid** and the **Embedded Structure**.

By selecting **Friction**, the bond slip behavior of the interface is defined by using a plasticity model. Enter a value for the **Cohesion** c_0 to define the initial resistance to sliding. You can also specify a **Hardening model** — **None**, **Linear**, or **User defined**.

- For **Linear**, enter the **Hardening coefficient** k_p. This option defines a linear hardening function $k_p u_{pe}$, where u_{pe} is the accumulated slip. The current sliding resistance is then $c = c_0 + k_p u_{pe}$.
- For **User defined**, enter an expression for the **Hardening function** c_h. The default expression is `0[unit]*<tag>.upe`. The unit depends on the **Embedded structure** interface, and the variable `<tag>.upe` is the accumulated slip. The current sliding resistance is then $c = c_0 + c_h$

The bond slip friction model formally describes the so-called *Tresca friction*, that is, the sliding resistance does not depend on the normal force acting on the interface between the **Solid** and the **Embedded Structure**. However, a Coulomb type friction model can by implemented by adding a dependence with respect to a "normal force" in the expression for the **Cohesion** c_0. The difficulty lies in estimating the normal force

ADVANCED

To display this section, click the **Show More Options** button ( ) and select **Advanced Physics Option** in the **Show More Options** dialog.

Enter a scalar positive value in the **Extrapolation tolerance**; the default is 0.3. This tolerance is used by an internal general extrusion operator that maps expressions from the **Solid** (source) to the **Embedded Structure** (destination). If a point of the embedded structure is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be within the solid. Otherwise, the mapping fails.

Select the **Calculate dissipated energy** checkbox as needed to compute the energy dissipated when including a **Bond slip model** in the connection.

- For more information about coupling different element types, see Coupling Techniques.
- For details about how work with embedded structures, see Modeling Embedded Structures and Reinforcements.
- For details about the formulation of this coupling, see Embedded Elements.

Beam Cross Section–Beam Coupling

Use the **Beam Cross Section–Beam Coupling** to transfer the cross section data from a Beam Cross Section interface to either a Beam Interface or a Pipe Mechanics interface. You need to use one such **Beam Cross Section–Beam Coupling** node for each type of cross section.

The **Beam Cross Section–Beam Coupling** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

BOUNDARY SELECTION

Select the boundaries that represent the current cross section in the Beam Cross Section interface. This means that the selection should be identical to the selection in a corresponding **Homogeneous Cross Section** node.

EDGE SELECTION, BEAM

Select the edges of the beams that have the intended cross section.

COUPLED INTERFACES

Select the **Beam Cross Section** interface to couple.

Select the **Beam** interface to couple. This can be either a Beam interface or a Pipe Mechanics Interface.

Beam–Beam Cross Section Coupling

Use the **Beam–Beam Cross Section Coupling** to transfer the results from a Beam Interface or a Pipe Mechanics interface to a Beam Cross Section interface for visualization. You can add more than one **Beam Cross Section–Beam Coupling** node, but they should have different selections of cross sections in the Beam Cross Section interface.

The location, the orientation of the cross section, and the direction of its extrusion along the beam axis are independent of how the actual beam is located.

This multiphysics coupling is only available in 3D.

The **Beam–Beam Cross Section Coupling** node is only available with some COMSOL products (see www.comsol.com/products/specifications/)

BOUNDARY SELECTION

Select the boundaries that represent the current cross section in the Beam Cross Section interface. This means that the selection should be identical to the selection in a corresponding **Homogeneous Cross Section** node.

EDGE SELECTION, BEAM

Select the edges of the beams, for which the stress plots are to be made. Normally, these beams should have the same cross section as in the selected boundaries. It is however only the section forces that are transferred to the Beam Cross Section interface. It is thus possible to view the stress distribution for any cross section, with a given set of section forces.

The selected edges must form a straight line. In the Beam Cross Section, the thickness property of the layered material must be set equal to the total length of the selected edges, in order for the results to be mapped.

BEAM ORIGIN

Select the point on the beam edge that should be treated as start point when mapping the results to the Beam Cross Section interface.

COUPLED INTERFACES

Select the **Beam Cross Section** interface to couple.

Select the **Beam** interface to couple. This can be either a Beam interface or a Pipe Mechanics Interface.

BEAM PROPERTIES

This section is related to the fact that the Beam Cross Section interface uses the layered technology. You will typically not need to change the settings.

ADVANCED

Enter an **Extrapolation tolerance**. This is the tolerance used in the extrusion coupling operator that maps between the physical beam and the axial coordinate in the Beam Cross Section interface.

18

Glossary

This Glossary of Terms contains finite element modeling terms in a structural mechanics context. For mathematical terms, and geometry and CAD terms specific to the COMSOL Multiphysics® software and documentation, see the glossary in the *COMSOL Multiphysics Reference Manual*. For references to more information about a term, see the index.

Glossary of Terms

anisotropy Variation of material properties with direction. Both global and local user-defined coordinate systems can be used to define anisotropic material properties.

arbitrary Lagrangian–Eulerian (ALE) method A technique to formulate equations in a mixed kinematic description. An ALE referential coordinate system is typically a mix between the material (Lagrangian) and spatial (Eulerian) coordinate systems.

augmented Lagrangian method A method for solving contact problems. Augmentation components are introduced for the contact pressure and the components of the friction traction vector. Additional iteration levels are added where the displacement, contact pressure and traction variables are solved separately. The algorithm repeats this procedure until it fulfills a convergence criterion.

axial symmetry Symmetry in both load and geometry, solves for the radial (r) and axial (z) displacement.

bar A line element that only has translational degrees of freedom, capable of sustaining axial forces, with no bending moments, torsional moments, or shear forces. Can be used on lines in 2D and 3D. Also called *spar* or *truss element*. In COMSOL Multiphysics the term *truss element* is used.

beam A line element having both translational and rotational degrees of freedom. Capable of sustaining axial forces, bending moments, torsional moments, and shear forces. Can be used on lines in 2D and 3D.

body forces Forces distributed through the volume of a body.

buckling The sudden collapse or reduction in stiffness of a structure under a critical combination of applied loads.

cable A tension-only truss member used to model large deformation including sag.

Cauchy stress The most fundamental stress measure defined as force/deformed area in fixed directions not following the body.

compliance matrix The inverse of the elasticity matrix. See *elasticity matrix*.

constitutive equations The equations formulating the stress–strain relationship of a material.

constraint Constrains the displacement or rotations to zero or a specified value.

contact model The mathematical method to model bodies that come into contact with each other.

contact pair A pair that consists of some *source boundaries* and *destination boundaries* and is used for contact modeling.

coordinate system Global Cartesian, local geometrical, application specific, and user-defined coordinate systems. Loads, constraints, material properties, and variables are defined in a specific coordinate system.

damping Dissipation of energy in a vibrating structure. A common assumption is viscous damping where the damping is proportional to the velocity. See also *Rayleigh damping*.

deformation gradient Tensor containing the complete information about the local straining and rotation of the material. It is a positive definite second rank tensor.

destination boundary One side of a *contact pair*; the *destination boundary* is prohibited to penetrate the *source boundary*.

double dogleg solver The default nonlinear solver for mechanical contact. This solver is also useful for highly nonlinear simulations such as large plastic deformation or hyperelastic materials.

eigenfrequency study Solving for the damped or undamped natural frequencies and vibration modes of a structure.

elasticity matrix The matrix D relating strain to stresses:

$$\sigma = D\varepsilon$$

equilibrium equation The equation expressing the equilibrium formulated in the *stress* components.

Eulerian Model described and solved in a coordinate system that is fixed (spatial frame). See also *Lagrangian* and *arbitrary Lagrangian-Eulerian method*.

first Piola–Kirchhoff stress A stress measure used when geometric nonlinearities arise. All forces in COMSOL Multiphysics in case of geometric nonlinearity are of this type.

flexibility matrix The inverse of the *elasticity matrix*. See *elasticity matrix*.

free vibration The undamped vibration of a structure after it is displaced from the equilibrium position and released. See also *eigenfrequency analysis*.

frequency response A harmonic analysis solving for the steady-state response from a harmonic excitation. Typically a frequency sweep is performed, solving for many excitation frequencies at one time.

geometric nonlinearity In solid mechanics, the deformation state characterized by finite (or large displacements) but small to moderate strains. Not all material models are suitable for large strain analysis, even though the displacement and rotation can be large.

Green–Lagrange strain Nonlinear strain measure used in large-deformation analysis. In a small strain, large rotation analysis, the *Green–Lagrange strain* corresponds to the engineering strain, with the strain values interpreted in the original directions. The *Green–Lagrange* strain is a natural choice when formulating a problem in the undeformed state. The conjugate stress is the *second Piola–Kirchhoff stress*.

initial strain The strain in a stress-free structure before it is loaded.

initial stress The stress in a nondeformed structure before it is loaded.

isotropic material A material where the material properties are independent of direction.

Lagrangian Model described and solved in a coordinate system that moves with the material. See also *Eulerian* and *arbitrary Lagrangian–Eulerian method*.

large deformation The deformations are so large so the nonlinear effect of the change in geometry or stress stiffening need to be accounted for. See also *geometric nonlinearity*.

linear buckling analysis Solves for the linear buckling load using the eigenvalue solver.

linear viscoelasticity See *viscoelastic material*.

load multiplier A load used in linearized buckling analysis for estimating the critical load at which a structure becomes unstable.

mass damping parameter Rayleigh damping parameter, the coefficient in front of the mass matrix.

mass participation factors A measure of the sensitivity of a certain eigenmode to a uniform acceleration.

mixed formulation A formulation used for nearly incompressible materials, where the mean stress have been added as a dependent variable to avoid numerical problems.

nonlinear geometry See *large deformations.*

orthotropic material An orthotropic material has at least two orthogonal planes of symmetry, where material properties are independent of direction within each plane. Such materials require nine elastic constants in the constitutive equations.

parametric study A study that finds the solution dependence due to the variation of a specific parameter.

pinned A constraint condition where the displacement degrees of freedom are fixed but the rotational degrees of freedom are free, typically used for frames modeled using beam and truss elements.

plane strain An assumption on the strain field where all out-of-plane strain components are assumed to be zero.

plane stress An assumption on the stress field, all out-of-plane stress components are assumed to be zero.

plate Thin plane structure loaded in the normal direction.

principle of virtual work States that the variation in internal strain energy is equal to the work done by external forces.

principal stresses/strains Normal stresses/strains with no shear components that act on the principal planes. The magnitude of the principal stresses/strains are independent of the coordinate system used.

rate of strain tensor The rate at which the strain tensor changes with respect to time in time-dependent studies.

quasistatic transient study The loads vary slow enough for the inertia terms to be negligible. A transient thermal study coupled with a structural analysis can often be treated as quasi static.

Rayleigh damping A viscous damping model where the damping is proportional to the mass and stiffness through the mass and stiffness damping parameters.

rotational degrees of freedom Degrees of freedom associated with a rotation around an axis. Beams, rigid connectors, rigid domains, and shells have *rotational degrees of freedom*.

second Piola–Kirchhoff stress Conjugate stress to *Green–Lagrange strain* used in large deformation analysis. The orientations of the stress components follow the material directions.

shell elements A thin element where both bending and membrane effects are included.

source boundary One side of a *contact pair*; the *destination boundary* is prohibited to penetrate the *source boundary*.

spar see *bar*.

spin tensor The skew-symmetric part of the velocity gradient tensor.

stationary study A study where the loads and constraints are constant in time. Also called *static*.

strain Relative change in length, a fundamental concept in structural mechanics.

stress Internal forces in the material, normal stresses are defined as forces/area normal to a plane, and shear stresses are defined as forces/area in the plane. A fundamental concept in structural mechanics.

stress stiffening The geometrically nonlinear effect which supplies the out-of-plane stiffness for membranes, for example.

stiffness damping parameter Rayleigh damping parameter, the coefficient in front of the stiffness matrix.

strain energy The energy stored by a structure as it deforms under load. Also called elastic energy.

time dependent study A time-dependent or *transient* study shows how the solution varies over time, taking into account mass, mass moment of inertia, and damping.

Tresca stress An equivalent stress measure that is equal to the maximum shear stress.

truss See *bar*.

viscoelastic material Viscoelastic materials have a time-dependent response, even if the loading is constant. Many polymers and biological tissues exhibit such a behavior. *Linear viscoelasticity* is a commonly used approximation where the stress depends linearly on the strain and its time derivatives.

Index

A

about layered shell 114
absolute values 681
absolute-tolerance parameters 450
abstract rigid domain 802
acoustic-structure interaction, frequency domain interface 321
activation 315
activation (node) 1372
  layered shell interface 1899
added mass (node) 1520
  layered shell interface 1946
  shell and plate interfaces 1814
added mass, interface (node) 1947
added mass, theory 886
adding connections 146
adhesion (node) 1599
adiabatic heating (node) 1526
adiabatic heating, layered (node) 1821
  membrane interface 2005
Anand viscoplasticity 737
angle of internal friction 648, 1329
angular acceleration and velocity 799
anisotropic materials
  defining 529
  elastic properties 1203, 1967
antisymmetry (node)
  beam interface 2053
  layered shell interface 1924
  membrane 2002
  shell and plate interfaces 1786
  solid mechanics 1442
  truss interface 2115, 2139
Application Libraries window 56
application library examples
  acpr interface 2328
  axisymmetric twist and bending 93, 97
beam cross sections 2079
beam end release 2071
beams 2019
beam–solid connection 151
bracket inertia relief 89, 439, 1454
Cam-clay material 1221
circumferential mode extension 1179, 1665, 1695
concrete 695
connecting shells and beams 130
cross-section data 2024
damping 1385
decohesion 292
dynamic_crack_branching 2352
eigenfrequency analysis 73, 112
external stress 1360
generation of time history from frequency domain results 68, 80
geometric nonlinearity 261
initial stress and strain 1357, 1993
large plastic strains 1286
layered shells 1864
layered shell-shell coupling 2369
linear buckling study 86, 403
load cases 161, 163
membrane interface 1961
MEMS materials database 195
modal mass 72
Mooney–Rivlin, two parameters 1242
multiple model method 2369
Ogden 1242
one-way FSI 338
piezoelectric devices 2266, 2274, 2294
piezoelectric materials database 195
pipe mechanics 2146
plasticity models 1286

poroelasticity 2244, 2252
portal crane 133
pressure acoustics, frequency domain 2328
prestressed bolts 86, 368
response spectrum analysis 87–88, 303, 428, 437
rigid connector 368, 1552, 1620, 1625, 1628
self-contact 303
sharing edges 2356, 2364
shell–beam connection 151
shells 1699
shell–solid connection 151
soil plasticity 1330
solid mechanics 1189
spring foundation 319
thermal bending of beam 2032
thermal expansion 1351
thermal expansion at constraints 332
thermal stress 2203
transient contact 298
trusses 2129
viscoelastic material 1279
viscoplastic material 1307
applied force (node) 1407, 1556
theory 804
applied moment (node) 1410, 1558
theory 804
applying
loads 159
moments 165
arbitrary Lagrangian-Eulerian (ALE) method 255
area scale factor, layered shell 1054
area, beam cross sections 1100
Arruda–Boyce material 602
associated flow rule 623
attachment (node)
layered shell interface 1953
shell and plate interfaces 1853
shells 2069
solid mechanics 1563
theory 803
attachments
theory 179
augmented Lagrangian method 42, 272, 912
average rotation (node) 1667
axial stress 1115
axial symmetry
constraints and 174
initial stress and strain 512
axisymmetric models
solid mechanics 93
azimuthal mode number 499
azimuthal wave number 894

B

back stress 665, 739
base excitation (node) 1451, 2164
Bauschinger effect 664
beam
coupling to a solid 148
beam cross section interface 2084
theory 1100
beam cross section–beam coupling (node) 2374
beam end release (node) 2070
beam interface 2016
theory 1069
beam-beam cross section coupling (node) 2375
beams
cross-section data 2023
initial stress and strain 1076
initial stresses and strains 2035
linear elastic material 2030

loads applied 2061
prescribed acceleration 2045
prescribed displacement/rotation 2040
prescribed velocity 2043
section orientation 2028
strain–displacement/rotation 1074
stress evaluation 1081
thermal expansion 2032
thermal strain 1075
bending shear stress 1115
bending shear stresses 1102
bending stress 1115
biaxial compression 678
biaxial data 680
biaxial tension 1339
bimoment 1114
Biot–Willis coefficient 2246, 2250, 2254
Blatz–Ko material 606
body load (node)
layered shell interface 1926
shell and plate interfaces 1787
solid mechanics 1446
bolt pretension (node) 1623
bolt pretension, theory 359
bolt selection (node) 1626
bolt thread contact (node) 1629
bolted joints 359
boundary conditions
contact pairs 283
shell and plate interfaces 1014
boundary load (node) 1460
for thin layer 1492
layered shell interface 1931
boundary nodes
beam interface 2020, 2147
layered shell 1865
membrane interface 1962
shell and plate interfaces 1699
solid mechanics 1189, 1685
truss interface 2094, 2124, 2129, 2171
boundary pair (node) 1534
boundary to boundary (node) 1824
bounds (node) 2173
box sections, beams 2023, 2096
Bresler–Pister criterion 677
buckling 40
built-in couplings 146
bulk modulus
elastic moduli 525, 1201, 1503

C

calcite 681
calculating stress and strain 1016
Cam-clay model 705
canonical systems 477
carbonate rocks 681
Cauchy stress 505
Cauchy stress tensor 253
Cayley–Hamilton theorem 489, 508
cell periodicity 1534
cell periodicity (node)
solid mechanics 1528
center of gravity, beam cross sections 1100
center of mass
boundary (node) 1413
edge (node) 1413
point (node) 1413
rigid domains 798
center of rotation 1104
attachments 181
boundary (node) 1553
edge (node) 1553
point (node) 1553
rigid domains 801
ceramics 678
change cross section (node)
solid mechanics 1198

change thickness (node)
solid mechanics 1197
charge conservation, piezoelectric (node) 1255
chemical potential (node)
transport in solids interface 2181
circle, Mohr 649
circular sections, beams 2023, 2096
circumferential mode number 499
Coble creep 728
coefficient of thermal expansion
beams 2033
shells and plates 1740
cohesion 648, 1329
cohesionless soils 652
cohesive zone model 290
cohesive–frictional materials 678
common sections, beam interface 2023
common sections, truss interface 2096
common settings 54
complex mechanical energy flux 1002
complex modulus 246
compressive meridians 509, 678–679
concrete 677, 679
concrete (node) 1335
conduction loss (time-harmonic) (node) 1391
connection
beam edge to shell edge 1028
beam point to shell boundary 1029
beam point to shell edge 1030
connections
beams, shells 1028
beams, shells, solids 1021
shell perpendicular to solid 1022
connections, adding 146
constitutive relation, membranes 1063
constraints 170
contact (node) 1572
contact formulation 297
continuity (node)
layered shell interface 1955
continuity, layered shell 1058
continuum mechanics forces 758
converse piezoelectric effect 209
coordinate system, beam cross sections 1102
coordinate systems
constraints and 172
loads and 159
local edge system 1776
solid mechanics theory 477
coordinate systems, membranes 1061
Coulomb friction 1593, 1619
coupling
beam to a shell 149
beam to a solid 148
membrane to a solid 147
shell to a solid 147
coupling loss (node) 1388
layered shell interface 1759, 1907
coupling operator 151
crack (node) 1631
crack closure (node) 1633
crack surface density function 1150
Craig–Bampton method 339
creep (node) 1298
layered shell interface 1882
membrane interface 1979
shell and plate interfaces 1727
creep dissipation rate density 748
creep strain rate 725
critical load factor 85, 399
cross-section data (node) 2023, 2096
crystal cleavage 681
crystal cut standards 212

crystals
elastic properties 1203
cyclic symmetry, theory 892
cylindrical coordinate systems 93

D damage (node) 1367
damped eigenfrequency study 245
damping
equation of motion, and 240
loss factors 244
losses and 239
point mass 2062, 2119
solid mechanics 1381
viscoelastic materials, and 579
damping (node)
layered shell interface 1904
membrane interface 1998
shell and plate interfaces 1756
solid mechanics 1380
damping models 244
decohesion (node) 1602
defining
anisotropic materials 529
constraints 170
isotropic materials 524
multiphysics models 320
orthotropic materials 527
thermoelastic materials 533
deformation gradient 480
deformation resistance 738
saturation coefficient 738
sensitivity 738
delamination (node) 1900
destination filter (node) 1571
destination point
boundary (node) 1518
edge (node) 1518
point (node) 1519
destination point (node) 1518
destinations and sources 278
deviatoric stress 633
dielectric loss (node) 1389
layered shell interface 1760, 1908
dielectric loss factor 225
diffusional creep 728
dilatational contributions 589
direct piezoelectric effect 209, 764
discretization, layered shell 1055
dislocation creep 728
dispersion curves 75
displacement field, defining 249
displacement gradients 480
displacement variables
element types and 145
dissipated energy 1001
dissipated energy density 630, 748
dissipated energy density rate 558
dissipation, piezoelectric materials 769
distortional contributions 589
disturbance factor 682
documentation 55
dolomite 681
domain nodes
solid mechanics 1189, 1685
double dogleg nonlinear solver 305
Dulong–Petit law 533
dynamic cyclic symmetry 893
dynamic frictional coefficients 926
dynamic substructure 339

E edge group (node) 2071
edge load (node)
beam interface 2054
layered shell interface 1933
shell and plate interfaces 1793
solid mechanics 1465
truss interface 2117, 2140
edge nodes

beam interface 2020, 2147
membrane interface 1962
solid mechanics 1189, 1685
truss interface 2094, 2124, 2129
edge selection (node) 1540, 1842
edge to boundary (node) 1827
edge to edge (node) 1829
effective stress 1116
effective stress tensor 682
eigenfrequency study 69
solid mechanics 832
eigenvalue solvers 69
eight-chain model 603
elastic deformation tensor 491
elastic domain 623
elastic energy 999
elastic Green–Lagrange strain tensor 492
elastic Hencky tensor 494
elastic material properties 1201–1203, 1965–1967
elastic moduli 524
elastic predeformation (node) 1652
elastic right Cauchy–Green tensor 492
elastic strain energy 611
elastic wire (node) 2131
elasticity matrix 524
electromechanical forces (node), multiphysics 2318
electromechanics interface 2257, 2263
theory 758, 774
electromigration (node) 2184
electrostatically actuated structures 2257, 2263
electrostriction (node) 2315
electrostriction multiphysics interface 2271
element types 145
emailing COMSOL 57
embedded reinforcement (node) 2370
energy dissipation 1001
energy function 584
energy quantities 999
equation of motion, damping and 240
equivalent plastic strain 625
equivalent single layer shell theory 1036
equivalent viscous damping 245
evanescent modes 75
excitation frequency 245
explicit damping 246
external chemical potential (node)
transport in solids interface 2183
external flux (node)
transport in solids interface 2184
external loads, shell and plate interfaces 1015
external strain (node) 1360
layered shell interface 1894
external stress (node) 1357, 2036, 2108
layered shell interface 1892
membrane interface 1993
shell and plate interfaces 1748
external stress–strain relation (node) 1392

F

face load (node) 1634
for thin layer 1488
layered shell interface 1928
shell and plate interfaces 1789
fasteners (node) 1535, 1836
ferroelectroelasticity multiphysics interface 2273
fiber (node) 1343
for thin layer 1480, 1985
shell and plate interfaces 1734
Filter 317
first order shear deformation laminate

theory 1037
first Piola–Kirchhoff stress 505
fixed constraint (node) 1403, 1429
for thin layer 1486
layered shell interface 1918
shell and plate interfaces 1764
fixed constraint, interface (node) 1919
fixed constraint, theory 803
fixed joint (node) 1569
Floquet periodicity, theory 892
flow rule 623
fluid and pipe materials (node) 2157
fluid load (node) 2167
fluid pore pressure 651
fluid–membrane interaction, fixed geometry 2224
fluid–pipe interaction 2343
fluid–pipe interaction, fixed geometry 2226
fluid–shell interaction 2218
fluid–solid interaction 2214, 2222, 2235
fluid–structure interaction 2216, 2322, 2339
fluid–structure interaction interface
theory 899
fluid–structure interaction, fixed geometry 2220
fluid–structure interaction, pair 2341
fold lines 1012
foldline connection, layered shell 1057
fold-line limit angle 1696, 1861
follower loads 1068
forces, continuum mechanics and 758
free (node) 1416
layered shell interface 1910
free-free modes 72
frequency domain study
solid mechanics 828
frequency domain, prestressed study 261
frequency response study
loss factor damping 245
friction (node) 1592, 1619
friction forces 1595
friction models 1593, 1619
friction, angle 648, 1329

G
Gao material 607
Garofalo law 727
general extrusion operator 151
generalized Hoek–Brown criterion 681
Generalized Kelvin–Voigt viscoelastic model 559
generalized Maxwell model 553
Gent material 604
geological strength index (GSI) 681
geometric nonlinearity 256
membranes 121
micromechanics, and 248
piezoelectric devices 261
geometrically linear formulation (checkbox) 256, 1204, 1213, 1226, 1249, 1258, 1265, 1393, 1497, 1708, 1713, 1722, 1763, 1875, 1968, 1973, 2032, 2039, 2100, 2159
glass transition temperature 577
global
coordinate systems 477
GMRES iterative solvers 449
gradient displacements 480
gravity (node) 1449
pipe mechanics interface 2163
Green–Lagrange strain 249
Green–Lagrange tensor 483

H
Haigh–Westergaard coordinates 509
hardening models
theory 659
harmonic loads 828

harmonic perturbation 83
harmonic perturbation (node)
shell and plate interfaces 1856
truss interface 1656, 2073
heat dissipation 246
Hencky plastic strain 626
Hencky strain 485
Hermitian matrices 448
Hill orthotropic plasticity 636
Hoek–Brown criterion 680
hole (node) 2087
homogeneous cross section (node) 2085
hoop stress 1130
H-profile sections, beams 2023, 2096
hydrostatic axis 509
hydrostatic pressure 506, 648
hygroscopic swelling 166
hygroscopic swelling (node) 1352
layered shell interface 1889
membrane interface 1989
hygroscopic swelling (node), multiphysics 2308
hyperbolic sinus 727
hyperelastic material 1000
hyperelastic material (node) 1231
layered shell interface 1870
hyperelastic material, layered (node) 1716
hyperelastic materials 595
nearly incompressible 594
theory 584
hysteretic loss 218

I

I.R.E. standard, for material orientation 210
ideal plasticity 660
IEEE standard, for material orientation 210
imperfection sensitivity 402
implementation
beams 1077
inelastic deformation tensor 492
inertia relief (node) 1452
inertial effects, contact modeling 297
inertial forces 799
initial stress and strain 771
theory 512
initial stress and strain (node) 1355
beam interface 2035
layered shell interface 1891
membrane interface 1991
shell and plate interfaces 1745
solid mechanics 1355
truss interface 2107, 2134
initial stress and strain, beams 1076
Initial Values (node) 2187
initial values (node)
beam interface 2022
layered shell interface 1867
rigid domains 1402
shell and plate interfaces 1703
solid mechanics 1196
initial yield stress 660
integration in a laminate 1038
intercalation strain (node) 1354
interior contact (node) 1585
internet resources 55
invariants 584
inverse piezoelectric effect 764
Ising spin model 793
isochoric
contributions 589
process 481
strain energy density 594
isochoric strain energy 611
isochoric-elastic
deformation 589

Green–Lagrange strain tensor 589
right Cauchy–Green tensor 589
isotropic materials
defining 524
elastic properties 1201, 1965
isotropic rocks 681
iterative solvers 448

J j-integral (node) 1635
joints, bolted 359
Joule heating and thermal expansion interface 321, 2210

K Kelvin–Voigt viscoelastic model 564
kinematic constraints 179
kinematic hardening 664
kinematics, rigid domain 796
kinetic energy 1001
knowledge base, COMSOL 57

L Lade–Duncan criterion 652
Lagrange elements 101
Lagrangian formulations 479
Lamé parameters 525, 1201
lamina constitutive law 1039
laminate constitutive law 1042
large deformation modeling 479
large deformations 41
piezoelectric materials 261
large strain plasticity 588
layered shell interface 1860
theory 1052
layered shell–shell connection (node) 2368
layered shell–structure cladding (node) 2366
left Cauchy–Green tensor 484
Lie derivative 624
limestone 681
limiting chain extensibility 604
line load (node) 1934
linear buckling study 40, 261
linear elastic material (node) 1198
beam interface 2030
layered shell interface 1868
membrane interface 1964
shell and plate interfaces 1706, 1715
truss interface 2100
linear elastic material, layered (node)
membrane interface 1970
shell interface 1709
linear elastic materials 189
linear viscoelasticity 550
linearized buckling analysis 85, 398
linearly accelerated frame (node) 1458, 2165
layered shell interface 1930
linper operator 83
load cases 161
load multiplier 398
loads
applied to beams 2061
pressure 165
singular 163
total 167
local coordinate systems 478
local edge system 1776
local system results (node) 1666
location
boundary (node) 1409
edge (node) 1409
point (node) 1409
locking 593
Lode angle 508
Hoek–Brown criterion 681
Willam–Warnke criterion 678
logarithmic decrement 240
logarithmic plastic strain 626

long-term shear modulus 553
loss factor damping
solid mechanics and 244
solid mechanics theory 244
springs, and 319
loss modulus 246, 579
losses and damping 239
low-reflecting boundary (node) 1534
low-reflecting boundary, theory 891

M macroscopic shear modulus 603
magnetic forces, rotating machinery 2331
magnetically actuated structures 2275
magnetomechanics 2324, 2328
magnetomechanics interface 2275
magnetomechanics, no currents interface 2281
magnetostatically actuated structures 2281
magnetostriction 778
marble 681
mass and moment of inertia (node) 1411, 1559
theory 802
mass density 481
mass moment of inertia 2061
material coordinates 476
material frame 249
material models 188
materials
nearly incompressible 190, 536
piezoelectric 764
viscoelastic 550
materials, hyperelastic 584
Matsuoka–Nakai criterion 650
max shear stress factor 1103
Maxwell stress tensor 759, 774
mechanical damping (node) 1386
layered shell interface 1906
shell interface 1758
mechanical energy flux 1002
membrane
coupling to a solid 147
membrane interface 1960
meridians, tensile and compressive 509, 678–679
Mindlin–Reissner type shell 109
MITC shell formulation 109, 1011
mixed formulations 192, 536
mobilized planes 651
mode analysis study 75
modeling, large deformations 479
modified Cam-clay model 705
modified tensors 590
Mohr–Coulomb criterion 648
moments 799
beams 2026, 2151
solid mechanics and 165
moments of inertia 123
moments of inertia, beam cross sections 1100
Mooney–Rivlin material
five parameters 597
nine parameters 598
two parameters 597
moving mesh interface, piezoelectric devices and 264
MPH files 56
mullins effect (node) 1280
layered shell interface 1878
multiaxial stress states 677
multibody dynamics interface 1563
multiphysics
electrostriction 2271, 2315
ferroelectroelasticity 2273
hygroscopic swelling 2308
nonlinear magnetostriction 2292, 2336

piezoelectric effect 2312
piezoelectricity 2264
piezomagnetic effect 2334
piezomagnetism 2289
poroelasticity 2242, 2247
multiphysics coupling
electromechanical forces 2318
fluid–membrane interaction, fixed geometry 2224
fluid–pipe interaction 2343
fluid–pipe interaction, fixed geometry 2226
fluid–shell interaction 2218
fluid–solid interaction 2214, 2222, 2235
fluid–structure interaction 2216, 2322, 2339
fluid–structure interaction, fixed geometry 2220
fluid–structure interaction, pair 2341
Joule heating and thermal expansion 2210
thermal expansion (node) 2299
thermal stress 2202
multiphysics modeling 320
MUMPS direct solvers 448
Murnaghan material 607

N Navarro–Herring creep 728
Navier–Stokes equations 899
nearly incompressible hyperelastic materials 594
nearly incompressible materials 190, 512, 536
Neo-Hookean material 595
Neumann boundary conditions
applied force 1407
applied moment 1410
No flux (node) 2187
no flux (node) 2176
no rotation (node)
beam interface 2050
shell and plate interfaces 1779
nodes, common settings 54
nominal stress 517
nonassociated flow rule 623
nonlinear elastic material 1000
nonlinear elastic material (node) 1209
nonlinear magnetostriction (node), multiphysics 2336
nonlinear magnetostriction multiphysics interface 2292
Norton equation 727

O octahedral plane 509
Ogden material 599
one-way model formulations 900
orientation, piezoelectric material 210
orthotropic materials
defining 527
elastic properties 1202, 1966
Ottosen criterion 679
over-consolidation pressure 711
overpressure 1130

P pair nodes
beam interface 2020, 2147
membrane interface 1962
solid mechanics 1189, 1685
truss interface 2094, 2124, 2129, 2171
parametric analysis 40
PARDISO direct solvers 448
penalty factors
contact node, and 1578, 1589
contact pairs and 284
penalty method 912
perfectly elastoplastic materials 660
perfectly plastic hardening 660
periodic boundary conditions 1523, 1528
periodic condition (node)

shell and plate interfaces 1817
solid mechanics 1523
periodic conditions, theory 892
phase (node) 1654
beam interface 2072
layered shell interface 1936
shell and plate interfaces 1855
solid mechanics 1654
phase field in solids interface 2170
phase field model (node) 2172
physics interfaces
connecting to DOF 145
physics interfaces, common settings 54
piezoelectric coupling 2264
piezoelectric crystal cut 210
piezoelectric effect (node) 2312
piezoelectric losses 218
piezoelectric material (node) 1255
layered shell 1719, 1872
piezoelectricity multiphysics interface 2264
piezomagnetic coupling 1264
piezomagnetic coupling matrices 779
piezomagnetic effect (node), multiphysics 2334
piezomagnetic material 1262
piezomagnetic material (node) 1262
piezomagnetism multiphysics interface 2289
pinned (node)
beam interface 2047
shell and plate interfaces 1777
truss interface 2111, 2134
pipe cross section (node) 2148
pipe mechanics interface 2144
theory 1129
pipe sections, beams 2023, 2096
pipes
stress evaluation 1138
plane stress and strain 91, 1178, 1180–1181
planes, symmetry and constraints 174
plastic multiplier 624
plastic potential 623
plasticity (node) 1281
layered shell interface 1879
membrane interface 1977
shell and plate interfaces 1725
plasticity models 1281
plate interface 1694
theory 1007
plates
external loads 1015
initial stresses and strains 1745
linear elastic material 1706
MITC shell formulation 1011
prescribed acceleration 1775
prescribed displacement/rotation 1766
prescribed velocity 1772
stress and strain calculations 1016
symmetry and antisymmetry 1014
point load (node) 1466
beam interface 2058
layered shell interface 1935
shell and plate interfaces 1799
point load on axis (node) 1476
point load, free (node) 1470
beam and pipe mechanics interfaces 2060
shell and plate interfaces 1803
point loads example 163
point mass (node)
beam interface 2061
shell and plate interfaces 1816
solid mechanics interface 1519
truss interface 2119

point mass damping (node)
beam interface 2062
shell and plate interfaces 1816
truss interface 2119
point nodes
beam interface 2020, 2147
membrane interface 1962
solid mechanics 1189, 1685
truss interface 2094, 2124, 2129
Poisson's ratio 190, 525, 536, 1201, 1503
polynomial hyperelastic material 597
poroelasticity multiphysics interface 2242, 2247
port (node) 1648
power law 727
Prager's hardening rule 665
Prandtl stress function 1111
predeformation (node) 1510
layered shell interface 1944
shell and plate interfaces 1813
prescribed acceleration (node) 1426
beam interface 2045
layered shell interface 1916
shell and plate interfaces 1775
prescribed acceleration, interface (node) 1917
prescribed chemical potential (node)
transport in solids interface 2188
prescribed displacement (node)
for thin layer 1483
layered shell interface 1911
solid mechanics 1417
prescribed displacement, interface (node) 1912
prescribed displacement, theory 803
prescribed displacement/rotation (node) 1404
beam interface 2040
shell and plate interfaces 1766
theory 803
prescribed phase field (node) 2174
prescribed transported quantity (node) 2189
prescribed velocity (node) 1424
beam interface 2043
layered shell interface 1913
shell and plate interfaces 1772
solid mechanics 1424
prescribed velocity, interface (node) 1915
pressure loads 165
pressure-wave speeds 525, 1201
prestressed analysis, eigenfrequency study 261
primary creep 725
principal stresses 507, 509
principal stretches 585
principle of virtual work 826
Prony series 552
propagating modes 75

Q quaternion constraint 797
quaternion representation, of rigid connector 905

R radius of gyration 1101
random vibration analysis 40
rate independent plasticity 621
rate of strain tensor 489
Rayleigh damping 242
rectangle sections, beams 2023, 2096
reduced component (node) 1568
reduced flexible components (node) 1567
reference coordinates 476
renaming
displacement DOF 146
representative volume element 356, 900

resonant frequency 242
response spectrum analysis 40, 429
results evaluation in a laminate 1049
results evaluation, for shells 1019
results evaluation, layered shells 117
reverse crack front (node) 1638
right Cauchy–Green deformation tensor 483, 584
right polar decomposition 482
right stretch tensor 482
rigid body 795
rigid connection type, shells 1023
rigid connector 905
rigid connector (node) 1543, 2062
  layered shell interface 1950
  shell and plate interfaces 1844
rigid connector theory 803
rigid connector, interface (node) 1951
rigid connectors
  kinematic constraints and 179
  moments and 165
rigid domain (node)
  layered shell interface 1909
  theory 795
rigid domains
  angular acceleration 799
  angular velocity 799
  dynamics 799
rigid material (node) 1398
  pipe mechanics interface 2159
rigid motion suppression (node) 1444
  layered shell interface 1925
ring load (node) 1473
  shell and plate interface 1805
ring load, free (node) 1474
  shell and plate interfaces 1807
rock mass 682
rock types 681
rocks 680
rocks (node) 1341
roller (node) 1434
  for thin layer 1487
  layered shell interface 1922
roller, interface (node) 1923
rotated coordinate system 213
rotating frame (node) 1454
  layered shell interface 1930
rotating machinery, magnetic-structure interaction 2287
rotation matrix 797
rotation, rigid body 796
rotational degrees of freedom 124
rotational joints, beams 179

S

safety (for fasteners) (node) 1539, 1841
safety (node) 806, 1374, 2089
  layered shell interface 1903
  membrane interface 1996
  shell and plate interfaces 1755
Saint-Venant torsion 1113, 1115
Saint-Venant's principle 163
Saint-Venant–Kirchhoff material 596
scaling of eigenvectors 69
second Piola–Kirchhoff stress 253, 478, 505
secondary creep 725
section forces (node) 1660
section orientation (node) 2028
  pipe mechanics interface 2152, 2156
section stiffness (node)
  shell and plate interfaces 1761
selecting
  solvers 447
serendipity elements 101
shape factors 680
shape memory alloy 750

shear area 1103
shear center 1104
shear correction factor computation 1047
shear modulus expression 525, 1201, 1503
shear stresses 507, 648
shear-wave speeds 525, 1201
shell
  coupling to a beam 149
  coupling to a solid 147
shell interface 1694
  theory 1007
shell–beam connection (node) 2362
shells
  external loads 1015
  initial stresses and strains 1745
  linear elastic material 1706
  MITC shell formulation 1011
  prescribed acceleration 1775
  prescribed displacement/rotation 1766
  prescribed velocity 1772
  stress and strain calculations 1016
  symmetry and antisymmetry 1014
shift function 1277
sign conventions 506
simply supported (node) 1780
singular loads 163
size factors 680
skew-symmetric part 489
slip velocity (node) 1597
slit boundary 360, 362
SLS model 565
soil deformation theory 711
soil plasticity (node) 1327
solid
  coupling to a beam 148
  coupling to a membrane 147
  coupling to a shell 147
solid (node) 2191
solid mechanics
  damping 1381
  edge loads 1465
  initial stresses and strains 1355
  prescribed acceleration 1426
  prescribed velocity 1424
solid mechanics interface 1176, 1680
  theory 475
solid–beaml connection (node) 2357
solid–thin structurel connection (node) 2353
solver methods
  augmented Lagrangian 912
  penalty method 912
solver parameters 447
solver settings 447
SOR line solvers 449
source (node) 2175
  transport in solids interface 2194
source filter (node) 1571
source point
  boundary (node) 1518
  edge (node) 1518
  point (node) 1518
source point (node) 1517
sources and destinations 278
spatial coordinates 476
spatial stress tensor 253
spatially mobilized planes (STP) 651
spin tensor 489
spot welds (node) 1832
spring constant 318
spring foundation 1414
spring foundation (node) 1414, 1500, 1561
  layered shell interface 1937

shell and plate interfaces 1809
spring foundation, interface (node) 1938
spring foundation, solid mechanics 318
spring foundation, theory 877
spring-damper (node) 1512
spring-damper material (node) 2110
stabilization (node)
membrane interface 2004
standard linear solid model 565
standard settings 54
static frictional coefficients 926
stationary solvers 447
steady-state creep 725
steady-state stiffness 553
storage modulus 246, 579
Storakers material 601
stored energy 1000
straight edge constraint (node) 2098
straight edges 1120
strain energy density 612, 1000
strain–displacement, trusses 1117
strain–displacement/rotation 1074
stress
Cachy 505
first Piola–Kirchhoff 505
second Piola–Kirchhoff 505
stress and strain, piezoelectric devices 234, 764, 779
stress components, beam cross sections 1105
stress evaluation, beams 1081
stress evaluation, pipes 1138
stress linearization (node) 1658
stress migration (node) 2185
stress stiffening 121
stress–strain relation
beams 1074
trusses 1118
structure–pipe connection (node) 2369
study steps, geometric nonlinearity and 256
study types
eigenfrequency 832
frequency domain, solid mechanics 828
parametric 40
surface traction and reaction forces 186
symmetric matrices 447
symmetry (node)
beam interface 2050
layered shell interface 1923
membrane interface 2000
phase field in solids interface 2176
shell and plate interfaces 1781
solid mechanics 1437
transport in solids interface 2194
truss interface 2113, 2137
symmetry constraints 174
symmetry plane (node) 1784
solid mechanics 1440

T

tangent modulus 660
tangential strains 1063
technical support, COMSOL 57
temperature loads 166
tensile meridian 509
tensile meridians 678–679
tertiary creep 725
theory
beam cross sections 1100
beam interface 1069
electromechanics 758, 774
fluid–structure interaction 899
layered shell interface 1052
pipe mechanics interface 1129
shell and plate interfaces 1007
solid mechanics interface 475
theory background of laminated shell

1033
thermal expansion
loads and 166
thermal expansion (for constraints) (node)
layered shell interface 1920
shell and plate interfaces 1771
thermal expansion (multiphysics coupling) 2299
thermal expansion (node) 1348
beam interface 2032
layered shell interface 1888
membrane interface 1987
pipe mechanics interface 2161
shell and plate interface 1738
solid mechanics 1348, 1432, 2048, 2112, 2135
thermal expansion, hyperelastic materials and 586
thermal strain
beams 1075
thermal stress interface 2202
modeling 320
thermal–electric–structural interaction 321
thermoelastic materials, defining 533
thermophoresis (node) 2186
thermorheologically simple viscoelastic materials 576
thickness and offset (node)
shell and plate interfaces 1703
thickness change (node) 1705
thin elastic layer (node) 1507
layered shell interface 1940
thin elastic layer, interface (node) 1943
thin elastic layer, solid mechanics 318
thin elastic layer, theory 878
thin layer (node) 1478
three-parameter model 565
torsion, beam cross sections 1111
torsional shear stress 1115
total Lagrangian formulation 41
total loads 167
t-profile sections, beams 2023, 2096
tractions 253
transient creep 725
translation, rigid body 796
Tresca effective stress variable 634
Tresca yield criterion 634
triaxial conditions 678
triaxial data 680
TRS material 577
true strain 485
true stress tensor 253
truss interface 2092, 2122
trusses
harmonic perturbation 1656, 2073
initial stresses and strains 2107, 2134
linear elastic material 2100
straight edge 1120
strain–displacement 1117
two-point tensor 481

U

uniaxial compression 677, 679, 681
uniaxial tension 633, 1336, 1339
units, loads and 159
u-profile sections, beams 2023, 2096
user defined material 610
using
coordinate systems 477
predefined variables 182
spatial and material coordinates 476
weak constraints 185
uspring variable 318

V

Varga material 602
variables
cross section data, beams 2080

deformation gradient tensor 482
density 481
elastic Green–Lagrange tensor 493
elastic right Cauchy–Green tensor 493
elastic, inelastic, and total volume ratio 493
equivalent plastic strain 660
Green–Lagrange tensor 484
invariants 611
isochoric elastic Green–Lagrange strain tensor 591
isochoric-elastic Cauchy–Green deformation tensor 590
isochoric-elastic right Cauchy–Green deformation tensor 590
material and spatial coordinates 477
predefined 182
principal elastic stretches 611
right Cauchy–Green deformation tensor 484
right stretch tensor 482
rotation tensor 482
thermal stretch and thermal volume ratio 586
Tresca effective stress 634
volumetric plastic strain 711
vdamper variable 319
Vicat softening temperature 578
Villari effect 778
virtual crack extension (node) 1638
viscoelastic materials
definition 190
frequency domain analysis and damping 579
temperature effects 576
theory 550
viscoelasticity (node) 1270
layered shell interface 1877
membrane interface 1976
shell and plate interfaces 1724
viscoplasticity (node) 1305, 1319
layered shell interface 1884
membrane interface 1982
shell and plate interfaces 1730
viscous damping 246, 319
Voigt form 211
Voigt notation 527
volume reference density 325
volume-preserving contributions 589
volumetric contributions 589
volumetric plastic strain variable 711
volumetric strain energy 611
volumetric strain energy density 594
von Mises criterion 633
von Mises effective stress 1116

W warpage (node) 1670
warping function 1111
warping, beam cross sections 1113
wave speeds 1202, 1966, 2031
wave speeds (node) 1672
waveguide 75
weak constraints, using 185
wear (node) 1605
websites, COMSOL 57
Weertman creep 728
Willam–Warnke criterion 678
wire interface 2128
WLF shift functions 577, 1277
wrinkling (node) 1984

Y yield function 622
yield stress levels 1282, 1292
yield surfaces 623
Young's modulus expression 525, 1201, 1503

Z Zener model 565