

# Structural Mechanics Module

User's Guide

# Structural Mechanics Module User's Guide

© 1998–2020 COMSOL

Protected by patents listed on [www.comsol.com/patents](http://www.comsol.com/patents), and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,457,932; 9,098,106; 9,146,652; 9,323,503; 9,372,673; 9,454,625; 10,019,544; 10,650,177; and 10,776,541. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement ([www.comsol.com/comsol-license-agreement](http://www.comsol.com/comsol-license-agreement)) 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](http://www.comsol.com/trademarks).

Version: COMSOL 5.6

## Contact Information

Visit the Contact COMSOL page at [www.comsol.com/contact](http://www.comsol.com/contact) to submit general inquiries, contact Technical Support, or search for an address and phone number. You can also visit the Worldwide Sales Offices page at [www.comsol.com/contact/offices](http://www.comsol.com/contact/offices) for address and contact information.

If you need to contact Support, an online request form is located at the COMSOL Access page at [www.comsol.com/support/case](http://www.comsol.com/support/case). Other useful links include:

- Support Center: [www.comsol.com/support](http://www.comsol.com/support)
- Product Download: [www.comsol.com/product-download](http://www.comsol.com/product-download)
- Product Updates: [www.comsol.com/support/updates](http://www.comsol.com/support/updates)
- COMSOL Blog: [www.comsol.com/blogs](http://www.comsol.com/blogs)
- Discussion Forum: [www.comsol.com/community](http://www.comsol.com/community)
- Events: [www.comsol.com/events](http://www.comsol.com/events)
- COMSOL Video Gallery: [www.comsol.com/video](http://www.comsol.com/video)
- Support Knowledge Base: [www.comsol.com/support/knowledgebase](http://www.comsol.com/support/knowledgebase)

Part number: CM021101

# C o n t e n t s

## Chapter 1: Introduction

<b>About the Structural Mechanics Module</b>	<b>28</b>
A Quick Overview of the Structural Mechanics Module . . . . .	28
What Problems Can It Solve? . . . . .	29
The Structural Mechanics Module Physics Interface Guide . . . . .	32
Common Physics Interface and Feature Settings and Nodes . . . . .	40
Where Do I Access the Documentation and Application Libraries? . . . . .	40
<b>Overview of the User's Guide</b>	<b>44</b>

## Chapter 2: Structural Mechanics Modeling

<b>Study Types</b>	<b>51</b>
Introduction . . . . .	51
Stationary Analysis . . . . .	51
Eigenfrequency Analysis . . . . .	52
Mode Analysis . . . . .	57
Time-Domain Analysis . . . . .	59
Frequency-Domain Analysis . . . . .	60
Mode Superposition . . . . .	62
Harmonic Perturbation . . . . .	64
Modal Reduced-Order Models . . . . .	67
Linearized Buckling Analysis . . . . .	72
Bolt Pretension Study . . . . .	74
Random Vibration (PSD) Study . . . . .	74
Response Spectrum Analysis Study . . . . .	75
<b>Selecting the Physics Interface</b>	<b>77</b>
Solid Mechanics . . . . .	77
Elastic Waves, Time Explicit . . . . .	80
Shell and Plate . . . . .	80

Membrane . . . . .	81
Beam . . . . .	81
Pipe Mechanics . . . . .	82
Truss . . . . .	82
<b>Selecting Discretization</b>	<b>83</b>
Shape Function Order . . . . .	83
Lagrange and Serendipity Shape Functions . . . . .	84
Choosing Shape Functions in Multiphysics Models . . . . .	84
Implicit Shape Function Orders . . . . .	85
<b>Coupling Different Element Types</b>	<b>87</b>
Introduction to the Element Types . . . . .	87
Coupling Techniques . . . . .	87
<b>Applying Loads</b>	<b>96</b>
Units, Orientation, and Visualization . . . . .	96
Load Cases . . . . .	98
Singular Loads . . . . .	100
Moments in the Solid Mechanics Interface . . . . .	101
Pressure . . . . .	102
Acceleration Loads . . . . .	102
Temperature Loads — Thermal Expansion . . . . .	102
Hygroscopic Swelling . . . . .	103
Total Loads . . . . .	103
Loads with a Spatial Variation . . . . .	104
Variables for Loads . . . . .	104
<b>Defining Constraints</b>	<b>106</b>
Rigid Body Motion . . . . .	106
Orientation . . . . .	107
Prescribed Displacements, Velocities, and Accelerations . . . . .	108
Symmetry Constraints . . . . .	109
Elemental and Nodal Constraints . . . . .	112
Suppressing Constraints on Lower Dimensions . . . . .	113
Kinematic Constraints . . . . .	115
Rotational Joints . . . . .	115

<b>Calculating Reaction Forces</b>	<b>116</b>
Using Predefined Variables to Evaluate Reaction Forces . . . . .	116
Using Weak Constraints to Evaluate Reaction Forces . . . . .	117
Using Surface Traction to Evaluate Reaction Forces. . . . .	118
Evaluating Surface Traction Forces on Interior Boundaries . . . . .	119
<b>Introduction to Material Models</b>	<b>120</b>
Material Models for Structural Mechanics . . . . .	120
Combination of Material Models . . . . .	122
Entering Material Data. . . . .	122
Introduction to Linear Elastic Materials . . . . .	123
Introduction to Viscoelastic Materials . . . . .	123
Mixed Formulation . . . . .	124
Modeling Damage . . . . .	124
About the Material Databases for the Structural Mechanics Module . . . . .	126
Using External Materials . . . . .	127
<b>Modeling Piezoelectric Problems</b>	<b>141</b>
About Piezoelectric Materials . . . . .	141
Piezoelectric Material Orientation . . . . .	142
Piezoelectric Losses. . . . .	150
References for Piezoelectric Damping . . . . .	155
Piezoelectric Coupling. . . . .	156
Create the Piezoelectric Effect Interface and Define Domains . . . . .	156
Complete Settings of Piezoelectric Materials . . . . .	158
Add Damping and Loss . . . . .	159
Define Material Properties . . . . .	159
Coupling Piezoelectricity with Acoustics . . . . .	160
<b>Modeling Electrostrictive and Ferroelectroelastic Materials</b>	<b>162</b>
Electrostriction Coupling. . . . .	162
Ferroelectricity . . . . .	162
Multiphysics Interfaces. . . . .	162
<b>Modeling Magnetostrictive Materials</b>	<b>164</b>
Magnetostriction Coupling . . . . .	164
Create the Magnetostriction Interface and Define Domains. . . . .	164
Linear vs. Nonlinear Magnetostriction . . . . .	166

<b>Mechanical Damping and Losses</b>	<b>167</b>
Damping Sources . . . . .	167
About Damping . . . . .	168
Rayleigh Damping . . . . .	170
Loss Factor Damping . . . . .	172
Viscous Damping . . . . .	173
Equivalence Between Loss Factor, Rayleigh, and Viscous Damping . . . . .	173
Explicit Complex Valued Damping . . . . .	174
Viscoelastic Damping . . . . .	174
Piezoelectric Damping . . . . .	174
Adding Damping in the Modal Solver . . . . .	174
<b>Modeling Geometric Nonlinearity</b>	<b>176</b>
Geometric Nonlinearity, Frames, and the ALE Method . . . . .	176
Geometric Nonlinearity for the Solid Mechanics Interface . . . . .	184
Geometric Nonlinearity for the Shell, Plate, Membrane, Beam and Truss Interfaces . . . . .	185
Solving Geometrically Nonlinear Problems . . . . .	185
Prestressed Structures. . . . .	186
Geometric Nonlinearity for the Piezoelectric Material. . . . .	189
References . . . . .	191
<b>Contact Modeling</b>	<b>192</b>
Overview. . . . .	192
Setting Up a Contact Problem . . . . .	193
Contact Pairs . . . . .	201
Meshing for Contact Analysis . . . . .	203
Settings for Contact Nodes . . . . .	204
Quasistatic Contact Analysis . . . . .	213
Dynamic Contact Analysis . . . . .	214
Multiphysics Contact Analysis . . . . .	216
Special Types of Contact Problems . . . . .	217
Fallback Nodes for Contact . . . . .	219
Solver Settings for Contact Analysis. . . . .	219
Monitoring the Solution . . . . .	223
Dependent Variables in Contact Analysis. . . . .	224
Important Contact Variables . . . . .	227
References for Contact Modeling . . . . .	229

<b>Activating and Deactivating Material</b>	<b>230</b>
<b>Springs and Dampers</b>	<b>233</b>
<b>Defining Multiphysics Models</b>	<b>235</b>
Thermal-Structure Interaction . . . . .	235
Acoustic-Structure Interaction. . . . .	236
Thermal-Electric-Structural Interaction . . . . .	236
<b>Thermally Coupled Problems</b>	<b>238</b>
Temperature-Dependent Material Data . . . . .	238
Thermal Expansion . . . . .	240
Constraints and Thermal Expansion. . . . .	246
Thermoelastic Damping . . . . .	247
<b>Fluid-Structure Interaction</b>	<b>249</b>
Deforming Fluid Domains . . . . .	249
Union or Assembly . . . . .	250
Contact and FSI . . . . .	251
One-Way FSI . . . . .	252
<b>Computing Mass Properties</b>	<b>254</b>
<b>Effective Properties of Periodic Structures</b>	<b>257</b>
RVE Modeling Using the Cell Periodicity Node . . . . .	257
<b>Pretensioned Bolts</b>	<b>260</b>
<b>Simplified Modeling of Bolt Threads</b>	<b>266</b>
<b>Modeling Embedded Structures and Reinforcements</b>	<b>268</b>
<b>Modeling Cracks</b>	<b>273</b>
The Crack Node . . . . .	273
J-integrals. . . . .	274
Stress Intensity Factors . . . . .	277

<b>Performing a Random Vibration Analysis</b>	<b>279</b>
<b>Performing a Response Spectrum Analysis</b>	<b>286</b>
<b>Stress Linearization</b>	<b>294</b>
<b>Solver Settings</b>	<b>300</b>
Symmetric Matrices . . . . .	300
Selecting Iterative Solvers . . . . .	301
Solver Settings for Viscoelasticity, Creep, and Viscoplasticity . . . . .	302
Specifying Tolerances and Scaling for the Solution Components . . . . .	303
Solver Settings . . . . .	303
<b>Result Presentation</b>	<b>305</b>
Local Orientations . . . . .	305
How an Expression Is Evaluated . . . . .	308
Quality Settings . . . . .	310
Gauss Point Evaluation. . . . .	311
Postprocessing of Eigenmodes . . . . .	312
Plotting Applied Loads . . . . .	314
<b>Part Libraries</b>	<b>318</b>

## Chapter 3: Structural Mechanics Theory

<b>Solid Mechanics Theory</b>	<b>323</b>
Introduction . . . . .	323
<b>Frames and Coordinate Systems</b>	<b>325</b>
Material and Spatial Coordinates . . . . .	325
Coordinate Systems. . . . .	326
<b>Analysis of Deformation</b>	<b>328</b>
Lagrangian Formulation . . . . .	328
Deformation Measures . . . . .	331
Invariants of Strain . . . . .	335



Inelastic Strain Contributions . . . . .	339
Axial Symmetry and Deformation . . . . .	342
Generalized Plane Strain . . . . .	345
Out-of-Plane and Circumferential Modes. . . . .	347
<b>Stresses</b>	<b>350</b>
Defining Stress. . . . .	350
Invariants of the Stress Tensor . . . . .	351
Plane Strain and Plane Stress Cases . . . . .	354
Initial Stresses and Strains . . . . .	355
External Stress. . . . .	356
Axial Symmetry and Stresses . . . . .	357
<b>Equation of Motion</b>	<b>359</b>
<b>Material Models</b>	<b>362</b>
Linear Elastic Material . . . . .	365
Mixed Formulation . . . . .	375
Nonlinear Elastic Materials . . . . .	377
Linear Viscoelastic Materials . . . . .	388
Large Strain Viscoelasticity . . . . .	414
Hyperelastic Material Models . . . . .	415
Elastoplastic Material Models . . . . .	438
Failure Criteria for Concrete, Rocks, and Other Brittle Materials . . . . .	472
Elastoplastic Soil Models . . . . .	479
Creep and Viscoplasticity. . . . .	494
Shape Memory Alloy . . . . .	508
Piezoelectric Material . . . . .	513
Magnetostrictive Material. . . . .	521
Electrostriction . . . . .	530
Ferroelectroelasticity . . . . .	532
Rigid Domain Model . . . . .	535
Damage Models . . . . .	545
Safety Factor Evaluation . . . . .	555
<b>Formulation of the Equilibrium Equations</b>	<b>576</b>
Equation Implementation. . . . .	576

<b>Study Types</b>	<b>578</b>
Frequency-Domain Studies . . . . .	578
Eigenfrequency Studies . . . . .	579
Random Vibration Theory . . . . .	582
Response Spectrum Analysis Theory . . . . .	589
Linear Buckling . . . . .	607
<b>Damping</b>	<b>609</b>
Rayleigh Damping Theory . . . . .	609
Loss Factor Damping Theory . . . . .	610
Viscous Damping Theory. . . . .	611
<b>Loads and Boundary Conditions</b>	<b>613</b>
Distributed Loads . . . . .	613
Rotating Frame and Gravity. . . . .	614
Spring Foundation and Thin Elastic Layer. . . . .	616
Spring-Damper Theory . . . . .	620
Added Mass. . . . .	626
Rigid Motion Suppression Boundary Condition . . . . .	626
Low-Reflecting Boundary Condition . . . . .	629
Cyclic Symmetry and Floquet Periodic Conditions . . . . .	630
Symmetry Condition with Translation. . . . .	631
Roller Boundary Condition . . . . .	632
Thermal Expansion of Constraints . . . . .	634
Fluid-Structure Interaction Theory . . . . .	636
Periodic Cell Theory . . . . .	638
Rigid Connector Theory . . . . .	641
Contact Analysis Theory . . . . .	644

<b>Stress Linearization</b>	<b>672</b>
<b>J-integral Theory</b>	<b>675</b>
<b>Embedded Elements</b>	<b>679</b>
<b>Average Rotation</b>	<b>683</b>
<b>Energy Quantities</b>	<b>685</b>
Elastic Energy . . . . .	685
Kinetic Energy . . . . .	687
Dissipated Energy . . . . .	687
Mechanical Energy Flux . . . . .	688
Energy Variables . . . . .	689
<b>References</b>	<b>692</b>

## Chapter 4: Solid Mechanics

<b>The Solid Mechanics Interface</b>	<b>702</b>
Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics . . . . .	707
Initial Values . . . . .	712
Change Thickness . . . . .	713
Linear Elastic Material . . . . .	713
Nonlinear Elastic Material . . . . .	720
Elastoplastic Soil Material. . . . .	726
Hyperelastic Material . . . . .	731
Shape Memory Alloy . . . . .	737
Phase Transformation Direction . . . . .	740
Piezoelectric Material . . . . .	741
Magnetostrictive Material. . . . .	744
Viscoelasticity . . . . .	746
Mullins Effect . . . . .	752
Plasticity . . . . .	753
Set Variables . . . . .	759
Creep . . . . .	762

Viscoplasticity . . . . .	765
Porous Plasticity . . . . .	771
Soil Plasticity . . . . .	775
Concrete. . . . .	778
Rocks . . . . .	779
Thermal Expansion (for Materials) . . . . .	780
Hygroscopic Swelling . . . . .	783
Initial Stress and Strain. . . . .	785
External Stress. . . . .	787
External Strain. . . . .	790
Damage . . . . .	794
Activation . . . . .	798
Safety . . . . .	800
Damping . . . . .	806
Mechanical Damping . . . . .	811
Coupling Loss . . . . .	813
Dielectric Loss. . . . .	813
Conduction Loss (Time-Harmonic) . . . . .	814
External Stress-Strain Relation. . . . .	815
Rigid Domain . . . . .	817
Initial Values (Rigid Domain) . . . . .	820
Fixed Constraint (Rigid Domain) . . . . .	821
Prescribed Displacement/Rotation . . . . .	822
Applied Force (Rigid Domain) . . . . .	825
Location Nodes . . . . .	827
Applied Moment (Rigid Domain) . . . . .	828
Mass and Moment of Inertia (Rigid Domain) . . . . .	829
Center of Mass Nodes. . . . .	830
Spring Foundation (Rigid Domain) . . . . .	831
Free. . . . .	833
Prescribed Displacement. . . . .	835
Prescribed Velocity . . . . .	838
Prescribed Acceleration . . . . .	840
Fixed Constraint . . . . .	843
Thermal Expansion (for Constraints) . . . . .	845
Roller . . . . .	847
Symmetry . . . . .	849
Symmetry Plane . . . . .	851

Antisymmetry . . . . .	853
Rigid Motion Suppression . . . . .	854
Body Load . . . . .	856
Gravity . . . . .	858
Rotating Frame . . . . .	860
Boundary Load . . . . .	863
Edge Load . . . . .	865
Point Load . . . . .	866
Ring Load . . . . .	867
Point Load (on Axis) . . . . .	868
Spring Foundation . . . . .	869
Thin Elastic Layer. . . . .	875
Predeformation . . . . .	879
Spring-Damper . . . . .	880
Source Point (for Spring-Damper) . . . . .	885
Destination Point (for Spring-Damper) . . . . .	885
Source Point Nodes. . . . .	886
Destination Point Nodes . . . . .	886
Added Mass. . . . .	886
Periodic Condition . . . . .	889
Cell Periodicity . . . . .	892
Boundary Pair . . . . .	895
Low-Reflecting Boundary. . . . .	896
Thin-Film Damping . . . . .	897
Rigid Connector . . . . .	898
Center of Rotation Nodes . . . . .	906
Thermal Expansion (Rigid Connector). . . . .	907
Applied Force (Rigid Connector). . . . .	908
Applied Moment (Rigid Connector). . . . .	910
Mass and Moment of Inertia (Rigid Connector) . . . . .	911
Spring Foundation (Rigid Connector) . . . . .	912
Attachment . . . . .	915
Thermal Expansion (Attachment). . . . .	916
Beam Connection . . . . .	918
Shell Connection. . . . .	919
Contact . . . . .	921
Friction . . . . .	932
Slip Velocity. . . . .	936

Adhesion . . . . .	939
Decohesion . . . . .	941
Wear . . . . .	943
Bolt Pretension . . . . .	946
Bolt Selection . . . . .	947
Bolt Thread Contact . . . . .	949
Thread Boundary Selection . . . . .	950
Crack . . . . .	951
Face Load . . . . .	953
J-Integral . . . . .	954
Port. . . . .	956
Elastic Predeformation. . . . .	960
Phase . . . . .	962
Harmonic Perturbation . . . . .	963
Stress Linearization . . . . .	965
Average Rotation. . . . .	966
Wave Speeds . . . . .	969
Important Variables in the Solid Mechanics Interface . . . . .	970

## Chapter 5: Shell and Plate

<b>Modeling with Plates and Shells</b>	<b>976</b>
Introduction . . . . .	976
Dependent Variables . . . . .	977
Layered and Nonlayered Shells . . . . .	979
Working with Section Stiffness . . . . .	980
Loads . . . . .	981
<b>Theory for the Shell and Plate Interfaces</b>	<b>983</b>
General Theory for the Shell and Plate Interfaces . . . . .	983
Theory for FSDT Laminated Shell . . . . .	1006
Theory for Hyperelastic Shells. . . . .	1021
Results Evaluation in a Laminate . . . . .	1021
Theory for Section Stiffness. . . . .	1022
References for the Shell Interface . . . . .	1023

<b>The Shell and Plate Interfaces</b>	<b>1024</b>
Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces . . . . .	1028
Initial Values . . . . .	1031
Thickness and Offset . . . . .	1032
Linear Elastic Material . . . . .	1034
Layered Linear Elastic Material. . . . .	1036
Shell Local System . . . . .	1039
Layered Hyperelastic Material . . . . .	1040
Viscoelasticity . . . . .	1043
Plasticity . . . . .	1044
Set Variables . . . . .	1045
Creep . . . . .	1046
Viscoplasticity . . . . .	1047
Thermal Expansion (for Materials) . . . . .	1049
Hygrosopic Swelling . . . . .	1052
Initial Stress and Strain. . . . .	1055
External Stress. . . . .	1058
External Strain. . . . .	1061
Damage . . . . .	1062
Mullins Effect . . . . .	1063
Safety . . . . .	1064
Damping . . . . .	1066
Section Stiffness . . . . .	1068
Fixed Constraint . . . . .	1070
Prescribed Displacement/Rotation . . . . .	1071
Thermal Expansion (for Constraints) . . . . .	1075
Prescribed Velocity . . . . .	1076
Prescribed Acceleration . . . . .	1079
Pinned . . . . .	1081
No Rotation . . . . .	1083
Simply Supported. . . . .	1084
Symmetry . . . . .	1085
Symmetry Plane . . . . .	1088
Antisymmetry . . . . .	1090
Body Load . . . . .	1091
Face Load . . . . .	1092
Edge Load . . . . .	1094

Point Load . . . . .	1096
Ring Load . . . . .	1098
Spring Foundation . . . . .	1099
Predeformation . . . . .	1103
Added Mass. . . . .	1103
Point Mass . . . . .	1105
Point Mass Damping . . . . .	1106
Periodic Condition . . . . .	1106
Solid Connection. . . . .	1111
Beam Connection . . . . .	1112
Rigid Connector . . . . .	1114
Thermal Expansion (Rigid Connector). . . . .	1121
Attachment . . . . .	1122
Thermal Expansion (Attachment). . . . .	1123
Phase . . . . .	1124
Harmonic Perturbation . . . . .	1125

## Chapter 6: Layered Shell

<b>Theory for the Layered Shell Interface</b>	<b>1128</b>
About Layered Shells . . . . .	1128
Theory Background for the Layered Shell Interface . . . . .	1129
<b>The Layered Shell Interface</b>	<b>1143</b>
The Shell Properties and Interface Selection Sections . . . . .	1146
Domain, Boundary, Edge, Point, and Pair Nodes for the Layered Shell Interface. . . . .	1148
Initial Values . . . . .	1153
Linear Elastic Material . . . . .	1154
Hyperelastic Material . . . . .	1156
Piezoelectric Material . . . . .	1158
Viscoelasticity . . . . .	1161
Mullins Effect . . . . .	1162
Plasticity . . . . .	1163
Set Variables . . . . .	1165
Creep . . . . .	1166



Viscoplasticity . . . . .	1167
Thermal Expansion (for Materials) . . . . .	1168
Hygroscopic Swelling . . . . .	1170
Initial Stress and Strain. . . . .	1171
External Stress. . . . .	1173
External Strain. . . . .	1174
Damage . . . . .	1175
Activation . . . . .	1177
Delamination . . . . .	1178
Safety . . . . .	1181
Damping . . . . .	1182
Mechanical Damping . . . . .	1184
Coupling Loss . . . . .	1185
Dielectric Loss. . . . .	1186
Rigid Domain . . . . .	1187
Contact . . . . .	1188
Free. . . . .	1189
Prescribed Displacement. . . . .	1190
Prescribed Displacement, Interface . . . . .	1191
Prescribed Velocity . . . . .	1192
Prescribed Velocity, Interface . . . . .	1193
Prescribed Acceleration . . . . .	1194
Prescribed Acceleration, Interface . . . . .	1196
Fixed Constraint . . . . .	1197
Fixed Constraint, Interface . . . . .	1198
Thermal Expansion (for Constraints) . . . . .	1199
Roller . . . . .	1200
Roller, Interface . . . . .	1201
Symmetry . . . . .	1202
Antisymmetry . . . . .	1203
Rigid Motion Suppression . . . . .	1204
Body Load . . . . .	1205
Face Load . . . . .	1207
Gravity . . . . .	1208
Rotating Frame . . . . .	1209
Boundary Load . . . . .	1210
Edge Load . . . . .	1212
Line Load. . . . .	1213

Point Load . . . . .	1214
Phase . . . . .	1215
Spring Foundation . . . . .	1216
Spring Foundation, Interface . . . . .	1217
Thin Elastic Layer. . . . .	1219
Thin Elastic Layer, Interface . . . . .	1220
Predeformation . . . . .	1221
Added Mass. . . . .	1222
Added Mass, Interface . . . . .	1223
Rigid Connector . . . . .	1224
Rigid Connector, Interface . . . . .	1226
Thermal Expansion (Rigid Connector). . . . .	1227
Attachment . . . . .	1228
Thermal Expansion (Attachment). . . . .	1229
Continuity . . . . .	1230

## Chapter 7: Membrane

<b>Theory for the Membrane Interface</b>	<b>1234</b>
About Membranes . . . . .	1234
Theory Background for the Membrane Interface . . . . .	1236
<b>The Membrane Interface</b>	<b>1241</b>
Boundary, Edge, Point, and Pair Nodes for the Membrane Interface. . .	1243
Thickness and Offset . . . . .	1246
Linear Elastic Material . . . . .	1247
Layered Linear Elastic Material. . . . .	1251
Viscoelasticity . . . . .	1254
Plasticity . . . . .	1255
Set Variables . . . . .	1256
Creep . . . . .	1257
Viscoplasticity . . . . .	1259
Thermal Expansion (for Materials) . . . . .	1260
Hygroscopic Swelling . . . . .	1262
Initial Stress and Strain. . . . .	1264
External Stress. . . . .	1266

Safety . . . . .	1268
Damping . . . . .	1270
Symmetry . . . . .	1272
Antisymmetry . . . . .	1274

## Chapter 8: Beam

<b>Modeling with Beams</b>	<b>1278</b>
About Beams . . . . .	1278
In-Plane Beams . . . . .	1278
3D Beam . . . . .	1279
Orientation of the Cross Section. . . . .	1280
Geometric Variables . . . . .	1281
Result Evaluation . . . . .	1282
Geometric Nonlinearity . . . . .	1283
Modeling Stiffeners . . . . .	1284
Nonrigid Joints . . . . .	1285
<b>Theory for the Beam Interface</b>	<b>1289</b>
Shape Functions . . . . .	1289
Geometric Nonlinearity . . . . .	1291
Strain-Displacement/Rotation Relation. . . . .	1294
Stress-Strain Relation . . . . .	1294
Thermal Strain. . . . .	1295
Hygroscopic Swelling . . . . .	1295
Initial Load and Strain . . . . .	1296
Implementation . . . . .	1297
Theory for Section Stiffness. . . . .	1299
Stress Evaluation . . . . .	1301
Limited Displacement . . . . .	1303
Common Cross Sections. . . . .	1304
<b>The Beam Interface</b>	<b>1322</b>
Boundary, Edge, Point, and Pair Nodes for the Beam Interface . . . . .	1325
Initial Values . . . . .	1327
Cross-Section Data . . . . .	1327

Section Orientation . . . . .	1332
Linear Elastic Material . . . . .	1335
Thermal Expansion (for Materials) . . . . .	1337
Hygroscopic Swelling . . . . .	1338
Initial Stress and Strain. . . . .	1339
External Stress. . . . .	1340
Section Stiffness . . . . .	1341
Prescribed Displacement/Rotation . . . . .	1344
Prescribed Velocity . . . . .	1347
Prescribed Acceleration . . . . .	1349
Pinned . . . . .	1351
Thermal Expansion (for Constraints) . . . . .	1352
No Rotation . . . . .	1353
Symmetry . . . . .	1354
Antisymmetry . . . . .	1356
Edge Load . . . . .	1358
Point Load . . . . .	1360
Point Mass . . . . .	1361
Point Mass Damping . . . . .	1362
Solid Connection. . . . .	1362
Shell Connection . . . . .	1363
Rigid Connector . . . . .	1364
Attachment . . . . .	1369
Beam End Release . . . . .	1370
Edge Group. . . . .	1371
Phase . . . . .	1372
Harmonic Perturbation . . . . .	1373

## Chapter 9: Beam Cross Section

<b>Using the Beam Cross Section Interface</b>	<b>1376</b>
About Beams and Cross-Section Data. . . . .	1376
Using the Beam Cross Section Interface . . . . .	1377
<b>Theory for the Beam Cross Section Interface</b>	<b>1382</b>
Cross-Section Properties. . . . .	1382

Computation of Stresses . . . . .	1396
-----------------------------------	------

**The Beam Cross Section Interface** **1399**

Hole . . . . .	1400
----------------	------

Safety . . . . .	1401
------------------	------

## Chapter 10: Truss

**Modeling with Truss Elements** **1404**

**Theory for the Truss Interface** **1406**

Theory Background for the Truss Interface . . . . .	1406
---	------

**The Truss Interface** **1415**

Boundary, Edge, Point, and Pair Nodes for the Truss Interface . . . . .	1417
---	------

Cross-Section Data . . . . .	1419
------------------------------	------

Straight Edge Constraint . . . . .	1420
------------------------------------	------

Linear Elastic Material . . . . .	1420
-----------------------------------	------

Set Variables . . . . .	1422
-------------------------	------

Shape Memory Alloy . . . . .	1423
------------------------------	------

Thermal Expansion (for Linear Elastic Material) . . . . .	1423
---	------

Thermal Expansion (for Shape Memory Alloy) . . . . .	1425
--	------

Hygroscopic Swelling . . . . .	1426
--------------------------------	------

Initial Stress and Strain. . . . .	1427
------------------------------------	------

External Stress. . . . .	1427
--------------------------	------

Spring-Damper Material . . . . .	1429
----------------------------------	------

Pinned . . . . .	1430
------------------	------

Thermal Expansion (for Constraints) . . . . .	1431
---	------

Symmetry . . . . .	1432
--------------------	------

Antisymmetry . . . . .	1434
------------------------	------

Edge Load . . . . .	1436
---------------------	------

Point Mass . . . . .	1437
----------------------	------

Point Mass Damping . . . . .	1437
------------------------------	------

Phase . . . . .	1438
-----------------	------

## Chapter 11: Pipe Mechanics

<b>Theory for the Pipe Mechanics Interface</b>	<b>1440</b>
About Pipes . . . . .	1440
Effects of Internal Pressure . . . . .	1441
Fluid Flow Forces. . . . .	1445
Effects of Thermal Expansion . . . . .	1447
Stress Evaluation . . . . .	1449
Connection Between Pipes and Structures . . . . .	1454
Cross Sections. . . . .	1456
<b>The Pipe Mechanics Interface</b>	<b>1461</b>
Boundary, Edge, Point, and Pair Nodes for the Pipe Mechanics Interface. . . . .	1464
Pipe Cross Section . . . . .	1465
Section Orientation. . . . .	1470
Fluid and Pipe Materials . . . . .	1472
Rigid Domain . . . . .	1474
Thermal Expansion (for Materials) . . . . .	1476
Gravity . . . . .	1478
Fluid Load . . . . .	1479

## Chapter 12: Multiphysics Interfaces and Couplings

<b>The Thermal Stress, Solid Interface</b>	<b>1486</b>
<b>The Thermal Stress, Shell Interface</b>	<b>1488</b>
<b>The Thermal Stress, Layered Shell Interface</b>	<b>1490</b>
<b>The Thermal Stress, Membrane Interface</b>	<b>1492</b>
<b>The Joule Heating and Thermal Expansion Interface</b>	<b>1494</b>
<b>Overview of Fluid-Structure Interaction Interfaces</b>	<b>1497</b>
<b>The Fluid-Solid Interaction Interface</b>	<b>1499</b>
<b>The Fluid-Solid Interaction, Fixed Geometry Interface</b>	<b>1502</b>
<b>The Fluid-Shell Interaction Interface</b>	<b>1504</b>
<b>The Fluid-Shell Interaction, Fixed Geometry Interface</b>	<b>1506</b>
<b>The Fluid-Membrane Interaction Interface</b>	<b>1508</b>
<b>The Fluid-Membrane Interaction, Fixed Geometry Interface</b>	<b>1510</b>
<b>The Fluid-Pipe Interaction, Fixed Geometry Interface</b>	<b>1512</b>
<b>The Fluid-Solid Interaction, Conjugate Heat Transfer Interface</b>	<b>1513</b>
<b>The Fluid-Solid Interaction, Two-Phase Flow, Phase Field Interface</b>	<b>1516</b>
<b>The Fluid-Solid Interaction, Two-Phase Flow, Phase Field,</b>	

<b>Fixed Geometry Interface</b>	<b>1518</b>
<b>The Piezoelectricity Interface</b>	<b>1521</b>
Coupling Piezoelectricity with Acoustics . . . . .	1524
<b>Piezoelectricity, Layered Shell Interface</b>	<b>1525</b>
<b>The Electrostriction Interface</b>	<b>1528</b>
<b>The Ferroelectroelasticity Interface</b>	<b>1529</b>
<b>The Magnetostriction Interface</b>	<b>1530</b>
<b>Hygroscopic Swelling Coupling</b>	<b>1533</b>
<b>Multiphysics Couplings</b>	<b>1534</b>
Thermal Expansion . . . . .	1534
Layered Thermal Expansion . . . . .	1537
Hygroscopic Swelling . . . . .	1540
Piezoelectric Effect . . . . .	1542
Layered Piezoelectric Effect . . . . .	1544
Electrostriction . . . . .	1545
Magnetostriction . . . . .	1549
Lorentz Coupling . . . . .	1550
Fluid-Structure Interaction . . . . .	1551
Fluid-Structure Interaction, Pair . . . . .	1554
Fluid-Structure Interaction, Fixed Geometry . . . . .	1556
Fluid-Pipe Interaction . . . . .	1556
<b>Couplings Between Structural Mechanics Interfaces</b>	<b>1559</b>
Solid-Thin Structure Connection . . . . .	1559
Solid-Beam Connection . . . . .	1562
Shell-Beam Connection . . . . .	1566
Layered Shell-Structure Transition . . . . .	1568
Layered Shell-Structure Cladding . . . . .	1569
Structure-Pipe Connection . . . . .	1570
Embedded Reinforcement . . . . .	1571



## Chapter 13: Glossary

**Glossary of Terms**

**1578**



# 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, 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 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 and cable elements
- 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 Material
- 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 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 analysis. 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 be taken into account.

### **TRANSIENT ANALYSIS**

A transient analysis finds the transient response for a time-dependent model, taking into account mass, 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 non-deterministic 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, 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. The Viscoelasticity materials are 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 interface supports contact with or without friction. The two contact algorithms available are based on the *augmented Lagrangian* and *penalty* 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 couple 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
<b>Acoustics</b>				
<b>Elastic Waves</b>				
Elastic Waves, Time Explicit		elte	3D, 2D, 2D axisymmetric	time dependent
<b>Fluid Flow</b>				
<b>Fluid-Structure Interaction</b>				
Fluid-Solid Interaction <sup>2</sup>		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Fluid-Solid Interaction, Fixed Geometry <sup>2</sup>		—	3D, 2D, 2D axisymmetric	stationary; stationary, one way; time dependent; time dependent, one way
Fluid-Shell Interaction <sup>2</sup>		—	3D, 2D axisymmetric	stationary; time dependent





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

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
 <b>Structural Mechanics</b>				
Solid Mechanics <sup>1</sup>		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; mode analysis; time dependent; time dependent, modal; time dependent, prestressed, 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
Thermal Stress, Solid <sup>2</sup>		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Thermal Stress, Shell <sup>2, 4</sup>		—	3D, 2D axisymmetric	stationary; time dependent
Thermal Stress, Membrane <sup>2, 4</sup>		—	3D, 2D axisymmetric	stationary; time dependent
Joule Heating and Thermal Expansion <sup>2</sup>		—	3D, 2D, 2D axisymmetric	stationary; time dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Shell		shell	3D	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, prestressed, 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, prestressed, 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	2D	stationary
Truss		truss	3D, 2D	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, prestressed, 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
Membrane		mbrn	3D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, prestressed, 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)
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)
Piezoelectricity <sup>2</sup>		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; time dependent; time dependent, modal; time dependent, prestressed, modal; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; small-signal analysis, frequency domain; linear buckling

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Piezoelectricity, Layered Shell <sup>2, 3, 5</sup>		—	3D	stationary; eigenfrequency; time dependent; frequency domain
Magnetostriction <sup>2,3</sup>		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
Ferroelectroelasticity <sup>2,9</sup>		—	3D, 2D, 2D axisymmetric	stationary; time dependent; frequency domain
Electrostriction <sup>2,9</sup>		—	3D, 2D, 2D axisymmetric	stationary; time dependent; frequency domain
<sup>1</sup> This physics interface is included with the core COMSOL package but has added functionality for this module.				
<sup>2</sup> This physics interface is a predefined multiphysics coupling that automatically adds all the physics interfaces and coupling features required.				
<sup>3</sup> Requires the addition of the AC/DC Module.				
<sup>4</sup> Requires the addition of the Heat Transfer Module				
<sup>5</sup> Requires the addition of the Composite Materials Module				
<sup>6</sup> Requires the addition of the CFD Module or the Microfluidics Module				
<sup>7</sup> Requires the addition of the Pipe Flow Module				
<sup>8</sup> Requires the addition of the Porous Media Flow Module				
<sup>9</sup> Requires the addition of the AC/DC Module or 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 internet 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 about 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*.

---

To open the **Help** window:

Win

- 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.


---

To open the **Help** window:

Mac

- In the **Model Builder** or **Physics Builder** click a node or window and then press F1.

Linux

- On the main toolbar, click the **Help** (  ) button.
- From the main menu, select **Help>Help**.

---

### *Opening the Documentation Window*

---

To open the **Documentation** window:


Win

- Press Ctrl+F1.
- From the **File** menu select **Help>Documentation** (  ).

---

To open the **Documentation** window:

Mac

- Press Ctrl+F1.
- On the main toolbar, click the **Documentation** (  ) button.

Linux

- 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.




[The Application Libraries Window](#) in the *COMSOL Multiphysics Reference Manual*.


---

### Opening the Application Libraries Window

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



- From the **Home** toolbar, **Windows** menu, click () **Applications Libraries**.
- From the **File** menu select **Application Libraries**.


To include the latest versions of model examples, from the **File>Help** menu, select () **Update COMSOL Application Library**.

---



Select **Application Libraries** from the main **File>** or **Windows>** menus.



To include the latest versions of model examples, from the **Help** menu select () **Update COMSOL Application Library**.

---

## CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at [info@comsol.com](mailto: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 are 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	<a href="http://www.comsol.com">www.comsol.com</a>
Contact COMSOL	<a href="http://www.comsol.com/contact">www.comsol.com/contact</a>
COMSOL Access	<a href="http://www.comsol.com/access">www.comsol.com/access</a>
Support Center	<a href="http://www.comsol.com/support">www.comsol.com/support</a>
Product Download	<a href="http://www.comsol.com/product-download">www.comsol.com/product-download</a>
Product Updates	<a href="http://www.comsol.com/support/updates">www.comsol.com/support/updates</a>
COMSOL Blog	<a href="http://www.comsol.com/blogs">www.comsol.com/blogs</a>
Discussion Forum	<a href="http://www.comsol.com/community">www.comsol.com/community</a>
Events	<a href="http://www.comsol.com/events">www.comsol.com/events</a>
COMSOL Application Gallery	<a href="http://www.comsol.com/models">www.comsol.com/models</a>
COMSOL Video Gallery	<a href="http://www.comsol.com/video">www.comsol.com/video</a>
Support Knowledge Base	<a href="http://www.comsol.com/support/knowledgebase">www.comsol.com/support/knowledgebase</a>

# 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 models, and 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 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 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 Interface](#) combines a Solid Mechanics interface with an Electrostatics interface. Piezoelectric materials in 3D, 2D plane strain and plane stress, and axial symmetry can be modeled.
- [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.
- [The Magnetostriction Interface](#) 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.





# 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 <https://www.comsol.com/products/specifications/>

In this chapter:

- Study Types
- Selecting the Physics Interface
- Selecting Discretization
- Coupling Different Element Types
- Applying Loads
- Defining Constraints
- Calculating Reaction Forces
- Introduction to Material Models
- Modeling Piezoelectric Problems
- 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
- Computing Mass Properties
- Effective Properties of Periodic Structures
- Pretensioned Bolts
- Simplified Modeling of Bolt Threads
- Modeling Embedded Structures and Reinforcements
- Modeling Cracks
- Performing a Random Vibration Analysis
- Performing a Response Spectrum Analysis
- Stress Linearization
- Solver Settings
- Result Presentation
- Part Libraries

# 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](#)
- [Modal Reduced-Order Models](#)
- [Linearized Buckling Analysis](#)
- [Bolt Pretension Study](#)
- [Random Vibration \(PSD\) Study](#)
- [Response Spectrum Analysis 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 have such time dependences.

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.

### CONSTRAINTS

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.



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,  $\lambda$ , 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. In case of damping, the eigenvalue equation is expanded to

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

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  $\mathbf{M}$  so that

$$\mathbf{u}_i^T \mathbf{M} \mathbf{u}_i = 1 \quad (2-2)$$

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.

#### 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 a 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.

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 it manually under **Definitions>Variable 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.

	<p>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.</p>
	<p>For an example showing how to compute modal participation factors and modal mass, see <i>In-Plane Framework with Discrete Mass and Mass Moment of Inertia</i>: Application Library path <b>Structural_Mechanics_Module/Verification_Examples/inplane_framework_freq.</b></p>
	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"> <li>• <a href="#">Eigenvalue Solver</a></li> <li>• <a href="#">Studies and Solvers</a></li> <li>• <a href="#">Postprocessing of Eigenmodes</a></li> <li>• <a href="#">Derived Values, Evaluation Groups, and Tables</a></li> </ul> <p>In the theory chapter of the <i>Structural Mechanics User's Guide</i>:</p> <ul style="list-style-type: none"> <li>• <a href="#">Modal Participation Factors</a></li> </ul>

It is possible to compute eigenfrequencies for structures which are not fully constrained; this is 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 can 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.

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.



## 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 as 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. The most common case is when there are viscoelastic materials in the structure. In this case, the eigenvalue equation [Equation 2-1](#) becomes

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

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

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


In order to get a correct solution to [Equation 2-3](#), the linearization point  $\omega_L$  must be close to the actual eigenvalue  $\omega$ . 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**.

---

## 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, with a generic name 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 the 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 and Circumferential Modes](#) 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 first law must be solved.
- When there are time dependencies in the material model, as for creep or viscoelasticity.

The most general way of doing that 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.

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 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 **Quasi static** in the physics interface settings.



In the *COMSOL Multiphysics Reference Manual*:

- [Time-Dependent Solver](#)
  - [Studies and Solvers](#)
-

## *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 linearization point. In such a case, the frequency-domain analysis can be considered as a very small perturbation around the linearization point.



### Harmonic Perturbation

---

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

- Add a Phase subnode to the load, in which you give the phase angle.
- Enter the load as a complex value, for example as  $100[\text{N}] * (1 + 0.5 * i) / \text{sqrt}(1.25)$ .

Most results of a frequency domain analysis are complex. In results evaluation, 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 following options:

- Plot  $u$  or  $\text{real}(u)$ . This gives the displacement at zero phase angle.
- Plot  $\text{imag}(u)$ . This gives the displacement at a phase angle of 90 degrees.
- Plot  $\text{abs}(u)$ . This gives the amplitude of the displacement.
- Plot  $\text{arg}(u)$ . This gives the phase angle of the displacement.

The reference phase, with respect to which the results above are reported.

Result quantities that are nonlinear in terms of the displacements, such as principal stresses, should be interpreted with great care. 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

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 frequencies of the load are 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](#)
- 

### 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 a linear perturbation around that state.
- There are no nonzero prescribed displacements.
- 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.

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

- Time Dependent, Modal
- Frequency Domain, Modal
- Time Dependent, Prestressed, 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 two latter have 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, 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](#)



For an example showing how to perform mode superposition in time and frequency domain, see *Various Analyses of an Elbow Bracket*: Application Library path **Structural\_Mechanics\_Module/Tutorials/elbow\_bracket**.

---

### *Harmonic Perturbation*

Analyses in the frequency domain assume that the problem your 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 loading 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 `Linear()` operator is applied to the value of the load.

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 STEP	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

Note the following:

- 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.

This frame split was introduced in version 5.3. As an effect, models created in an earlier version, in which some expressions have a frame dependency, may produce results that differ from before. Examples of such cases are:

- Pressure loads
- Loads defined in coordinate systems with deformation dependent axis orientation
- User-defined expressions containing spatial (“lowercase”) coordinates



In the *COMSOL Multiphysics Reference Manual*:

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

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.



---

For an example showing how to use harmonic perturbation, see *Bracket*  
— *Frequency-Response Analysis*: Application Library path  
**Structural\_Mechanics\_Module/Tutorials/bracket\_frequency**.

---

### *Modal Reduced-Order Models*

---

Reduced-order modeling seeks to reduce the number of degrees of freedom in a physical model whilst still retaining the essential physics. For a lightly damped resonant system driven at one of its resonant frequencies, it is reasonable to consider only the contributions to the system of a small number ( $m$ ) of modes within the signal bandwidth. In some cases, a single mode is sufficient. A system with  $n$  degrees of freedom has mass, stiffness, and damping matrices of size  $n$ -by- $n$ . A reduced-order representation of the system considering  $m$  modes has size  $m$ -by- $m$ . The reduction in complexity of the system, and the computational speed up is therefore significant when  $m \ll n$ . This section describes the theory of the reduced-order system and gives guidelines on how to obtain reduced-order models from a COMSOL model.

This can be employed in two different ways: Either you can use the built-in modal solvers for the time or frequency domain, or you can export the small equivalent system and analyze it outside COMSOL Multiphysics, for example, as a component in a larger system simulation.

#### **THE MODAL COORDINATE SYSTEM**

Consider a mechanical system, with  $n$  degrees of freedom, described by an equation of the form

$$M\ddot{\mathbf{u}} + D\dot{\mathbf{u}} + K\mathbf{u} = \mathbf{F} \quad (2-5)$$

where  $\mathbf{u}$  is the displacement vector (size:  $n$ -by-1),  $K$  is the stiffness matrix (size:  $n$ -by- $n$ ),  $D$  is the damping matrix (size:  $n$ -by- $n$ ), and  $M$  is the mass matrix (size:  $n$ -by- $n$ ). In the frequency domain the problem takes the form

$$-\omega^2 M\mathbf{u}_0 + i\omega D\mathbf{u}_0 + K\mathbf{u}_0 = \mathbf{F}$$

where  $\mathbf{u} = \mathbf{u}_0 e^{i\omega t}$ .

Initially consider the system in the absence of damping and forces. The undamped system has  $n$  eigenvalues  $\omega_i$ , which satisfy the equation

$$K\hat{\mathbf{u}}_i = \omega_i^2 M\hat{\mathbf{u}}_i \quad (2-6)$$

These eigenvectors can be shown to be orthogonal with respect to both  $M$  and  $K$ :

$$\hat{\mathbf{u}}_j^T M\hat{\mathbf{u}}_i = 0 \quad i \neq j, \omega_i \neq \omega_j \quad (2-7)$$

$$\hat{\mathbf{u}}_j^T K\hat{\mathbf{u}}_i = 0 \quad i \neq j, \omega_i \neq \omega_j \quad (2-8)$$

Next the following  $n$ -by- $n$  matrix is constructed, with columns taken from the  $n$  eigenvectors:

$$U = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2 \dots \hat{\mathbf{u}}_n]$$

Then consider the following matrix:

$$U^T M U = \begin{bmatrix} \hat{\mathbf{u}}_1^T M\hat{\mathbf{u}}_1 & \hat{\mathbf{u}}_1^T M\hat{\mathbf{u}}_2 & & & & & \\ \hat{\mathbf{u}}_2^T M\hat{\mathbf{u}}_1 & \hat{\mathbf{u}}_2^T M\hat{\mathbf{u}}_2 & & & & & \\ & & \ddots & & & & \\ & & & \hat{\mathbf{u}}_{n-1}^T M\hat{\mathbf{u}}_{n-1} & \hat{\mathbf{u}}_{n-1}^T M\hat{\mathbf{u}}_n & & \\ & & & \hat{\mathbf{u}}_n^T M\hat{\mathbf{u}}_{n-1} & \hat{\mathbf{u}}_n^T M\hat{\mathbf{u}}_n & & \end{bmatrix}$$

From Equation 2-7 it is clear that this is a diagonal matrix. Similarly, from Equation 2-8 it is clear that  $U^T K U$  is also diagonal.

From the properties of the eigenvectors it is possible to expand any function in terms of the eigenvectors. Thus, the displacement  $\mathbf{u}$  can be written as:

$$\mathbf{u} = \sum_{i=1}^n a_i \hat{\mathbf{u}}_i$$

This equation can also be expressed in the form:

$$\mathbf{u} = U \mathbf{a} \quad (2-9)$$

where  $\mathbf{a}$  is a column vector containing the coefficients  $a_i$  as rows. In general  $\mathbf{a}$  is time-dependent.

Now consider the original equation: Equation 2-5. First substitute for  $\mathbf{u}$  using Equation 2-9. Then transform the equation to the *modal coordinate system* by premultiplying by  $U^T$ . This gives:

$$U^T M U \ddot{\mathbf{a}} + U^T D U \dot{\mathbf{a}} + U^T K U \mathbf{a} = U^T \mathbf{F} \quad (2-10)$$

It has already been established that the matrices  $U^T M U$  and  $U^T K U$  are diagonal and frequently a damping model is chosen that results in a diagonal damping matrix. For example, in Rayleigh damping  $D = \alpha M + \beta K$ , where  $\alpha$  and  $\beta$  are constants. For a general damping, the transformed damping matrix is however not diagonal. As an alternative, a damping ratio,  $\zeta_i$ , can be assigned to each mode.

### EIGENVALUE SCALING

The precise form of Equation 2-10 is determined by the normalization adopted for the eigenfunctions. In structural applications the eigenfunctions are often normalized such that  $U^T M U = I$ . This is referred to as *mass matrix* scaling in the eigenvalue solver. In this case Equation 2-6 gives

$$\hat{\mathbf{u}}_i^T K \hat{\mathbf{u}}_i = \omega_i^2 \hat{\mathbf{u}}_i^T M \hat{\mathbf{u}}_i = \omega_i^2$$

so that

$$U^T K U = \text{diag}(\omega_i^2)$$

where  $\text{diag}(\omega_i^2)$  is the diagonal matrix with diagonal elements  $\omega_i^2$ . Similarly, if damping ratios for each mode are defined, the damping matrix can be expressed in the form

$$U^T D U = \text{diag}(2\zeta_i \omega_i)$$

Thus, if *mass matrix* scaling is used Equation 2-10 takes the form

$$\ddot{\mathbf{a}} + \text{diag}(2\zeta_i \omega_i) \dot{\mathbf{a}} + \text{diag}(\omega_i^2) \mathbf{a} = U^T \mathbf{F} \quad (2-11)$$

It is also possible to scale the eigenvectors so that the point of maximum displacement has given displacement. This is referred to as *max* scaling in the eigenvalue solver. For an individual mode this scaling has a simple physical interpretation — the corresponding component of  $\mathbf{a}$ ,  $a_i$ , is the amplitude of the  $i$ :th mode, measured at the

point of maximum displacement, when the mode is driven by the force  $\mathbf{F}$ . In this case Equation 2-10 takes the form

$$\text{diag}(m_{\text{eff}, i})\ddot{\mathbf{a}} + \text{diag}(c_{\text{eff}, i})\dot{\mathbf{a}} + \text{diag}(k_{\text{eff}, i})\mathbf{a} = U^T \mathbf{F} \quad (2-12)$$

Here  $m_{\text{eff}, i}$  is the effective mass of the  $i$ :th mode,  $c_{\text{eff}, i} = 2m_{\text{eff}, i}\xi_i\omega_i$  is the effective damping parameter for the mode, and  $k_{\text{eff}, i}$  is the effective spring constant. Each element of the vector  $U^T \mathbf{F}$  gives the force component that acts on each of the respective modes.



When using max scaling, it is the largest value of a degree of freedom which is scaled to 1. The total displacement in that node will thus be between 1 and  $\sqrt{3}$  times the given scaling factor.

If degrees of freedom other than displacements are active in the eigenfrequency problem, the maximum value may occur in another type of degree of freedom such as electric potential or pressure. Consequently, the peak displacement in that mode can then be less than the scaling factor.

### REDUCED-ORDER MODELS

The preceding discussion did not consider how to reduce the number of degrees of freedom in the system. For systems in which the vector  $U^T \mathbf{F}$  has only a few significant components (for example, components  $i = 1, \dots, m$  where  $m \ll n$ ) the following approximation can be made:

$$\mathbf{u} \approx \sum_{i=1}^m a_i \hat{\mathbf{u}}_i$$

The expression for  $\mathbf{u}$  in matrix becomes:

$$\mathbf{u} = U \mathbf{a}'$$

where  $U'$  is now an  $m$ -by- $n$  and  $\mathbf{a}'$  is a vector with  $m$  components. The equation system in modal coordinates now takes the form

$$U^T M U \ddot{\mathbf{a}}' + U^T D U \dot{\mathbf{a}}' + U^T K U \mathbf{a}' = U^T \mathbf{F} \quad (2-13)$$

The matrices  $U^T M U$ ,  $U^T D U$ , and  $U^T K U$  now have dimensions  $m$ -by- $m$ . Similarly, the vector  $U^T \mathbf{F}$  has  $m$  components. This results in a significant reduction in the system complexity.

### REDUCED-ORDER MODELS WITH PHYSICAL DAMPING

If physically relevant damping is present in the system, the above theory must be modified as the damping matrix is no longer diagonal in the modal coordinate system. COMSOL can still handle this case as the modal solver does not assume that any of the matrices are diagonal. In this case the eigenvalues become complex and the eigenvectors split into right and left eigenvectors. The right eigenvectors  $U^r$  are solutions of the equation:

$$-\omega_i^2 \hat{M} \hat{\mathbf{u}}_{r,i} + i \omega_i \hat{D} \hat{\mathbf{u}}_{r,i} + K \hat{\mathbf{u}}_{r,i} = \mathbf{F}$$

As in the previous section, for a reduced set of modes, it is assumed that:

$$\mathbf{u} = U^r \mathbf{a}'$$

where  $U^r$  is the  $n$ -by- $m$  matrix containing the right eigenvectors chosen for the modal analysis. Once again  $\mathbf{a}'$  is a vector with  $m$  components. The system in modal coordinates takes the form

$$U_l^T M U^r \ddot{\mathbf{a}}' + U_l^T D U^r \dot{\mathbf{a}}' + U_l^T K U^r \mathbf{a}' = U_l^T \mathbf{F}$$

where  $U_l$  is the  $n$ -by- $m$  matrix containing the left eigenvectors chosen for the modal analysis.

The matrices  $U_l^T M U^r$ ,  $U_l^T D U^r$ , and  $U_l^T K U^r$  are no longer necessarily diagonal. The modal solver accepts any linearly independent set of vectors to project the solution vector and equations onto and constructs the reduced-order system accordingly.

### ACCESSING THE REDUCED-ORDER MODEL MATRICES

The **Model Reduction** and **Modal Reduced Order Model** study steps have the property that they can assemble the modal matrices and make them available for output. In the **Model Reduction** node, the **Store reduced matrices** check box must be selected.

After the model has solved, right-click the **Results>Derived Values** node and select **System Matrices**. In the output section choose the **Matrix** to display in the list. The mass matrix corresponds to the matrix  $U_l^T M U^r$ , the stiffness matrix corresponds to  $U_l^T K U^r$ , and the damping matrix corresponds to  $U_l^T D U^r$ . The vector  $U_l^T \mathbf{F}$  is available as the load vector. These matrices are given in a format that respects the

normalization of the preceding **Eigenvalue Solver**. To change this, select the **Eigenvalue Solver** node, and change the **Scaling of Eigenvectors** setting under the **Output** section. Use the **Max** setting if an equivalent Mass-Spring-Damper system is required, in which case the modal amplitude corresponds to the maximum displacement of the mode.

### *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  $\mathbf{f}$ ,

$$\mathbf{K}\mathbf{u} = (\mathbf{K}_L + \mathbf{K}_{NL})\mathbf{u} = \mathbf{f}$$

Here the total stiffness matrix,  $\mathbf{K}$ , has been split into a linear part,  $\mathbf{K}_L$ , and a nonlinear contribution,  $\mathbf{K}_{NL}$ .

In a first order approximation,  $\mathbf{K}_{NL}$  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  $\mathbf{f}_0$ ,

$$\mathbf{K}_L\mathbf{u}_0 = \mathbf{f}_0 \tag{2-14}$$

then the nonlinear problem can be approximated as

$$(\mathbf{K}_L + \lambda\mathbf{K}_{NL}(\mathbf{u}_0))\mathbf{u} = \lambda\mathbf{f}_0$$

where  $\lambda$  is called the *load multiplier*.

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  $\lambda$ .

$$(\mathbf{K}_L + \lambda\mathbf{K}_{NL}(\mathbf{u}_0))\mathbf{u} = 0$$

COMSOL reports a *critical load factor*, which is the value of  $\lambda$  at which the structure becomes unstable. The corresponding deformation is the shape of the structure in its buckled state.



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

### 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 2-14](#), but rather

$$(\mathbf{K}_L + \mathbf{K}_{NL}(\mathbf{u}_0))\mathbf{u}_0 = \mathbf{f}_0 \quad (2-15)$$

The assumption for the buckling analysis is still that  $\mathbf{K}_{NL}$  is proportional to the external load, even though this may be disputable for a strongly nonlinear case.  $\mathbf{K}_{NL}$  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 2-15](#), and the eigenvalue problem is reformulated as

$$((\mathbf{K}_L + \mathbf{K}_{NL}(\mathbf{u}_0)) + (\lambda - 1)\mathbf{K}_{NL}(\mathbf{u}_0))\mathbf{u} = 0$$



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 the critical load factor  $\lambda$  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.





For a structure that exhibits axial symmetry in the geometry, constraints, and loads, the critical buckling mode shape can still be nonaxisymmetric. A full 3D model should always be used when computing buckling loads.

	<ul style="list-style-type: none"> <li>• <a href="#">Studies and Solvers</a> and <a href="#">Linear Buckling</a> in the <i>COMSOL Multiphysics Reference Manual</i></li> <li>• <a href="#">Linear Buckling</a> in the theory section of the <i>Structural Mechanics Module User's Guide</i></li> </ul>
	<ul style="list-style-type: none"> <li>• <i>Bracket — Linear Buckling Analysis</i>: Application Library path <b>Structural_Mechanics_Module/Tutorials/bracket_linear_buckling</b></li> <li>• <i>Linear Buckling Analysis of a Truss Tower</i>: Application Library path <b>Structural_Mechanics_Module/Buckling/truss_tower_buckling</b></li> </ul>

### *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.

	<a href="#">Pretensioned Bolts</a>
	<i>Prestressed Bolts in a Tube Connection</i> : Application Library path <b>Structural_Mechanics_Module/Contact_and_Friction/tube_connection</b>

### *Random Vibration (PSD) Study*

Random vibration study is used to analyze the response of a structure subjected to loads that randomly vary in time, but 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 three studies and a number of nodes under **Global Definitions** added to the model.

---

	<ul style="list-style-type: none"><li>• See <a href="#">Performing a Random Vibration Analysis</a> for a detailed description of how to study random vibration.</li><li>• The theory is described in <a href="#">Random Vibration Theory</a>.</li></ul>
	<ul style="list-style-type: none"><li>• <i>Bracket — Random Vibration Analysis</i>: Application Library path <b>Structural_Mechanics_Module/Tutorials/bracket_random_vibration</b></li><li>• <i>Random Vibration Analysis of a Deep Beam</i>: Application Library path <b>Structural_Mechanics_Module/Verification_Examples/random_vibration_deep_beam</b></li></ul>

---

### *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 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 missing mass correction, you need also to compute a set of stationary load cases. To set up that analysis, use the **Missing mass correction study: Create** button 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**

# 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 objects with extension in all directions
- [Elastic Waves, Time Explicit](#) — For efficient analysis of linear elastic wave propagation in solid domain.
- [Shell and Plate](#) — For objects which are thin in one direction, but have significant bending stiffness
- [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
- [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



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

---

## *Solid Mechanics*

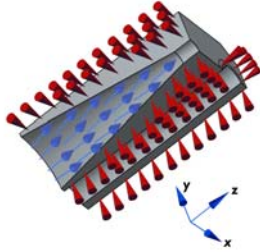
---

The Solid Mechanics interface offers the most general modeling of structural mechanics problems and is formulated based on general principles of continuum mechanics. It is the interface which contains the largest number of material models, and the most advanced boundary conditions. It is also the only physics interface that supports contact analysis.

The drawback with using solid elements 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.

### 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.

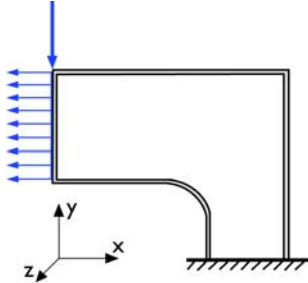


*Figure 2-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.*

### 2D GEOMETRY

#### Plane Stress

The plane stress variant of the 2D physics interface 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.*

The 2D physics interface for plane stress allows loads in the  $x$  and  $y$  directions, and it 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 flat plate in the  $xy$ -plane loaded only in its own plane and without any  $z$  direction restraint.

### Plane Strain

The plane strain variant of the 2D physics interface that assumes that all out-of-plane strain components of the total strain  $\epsilon_z$ ,  $\epsilon_{yz}$ , and  $\epsilon_{xz}$  are zero.

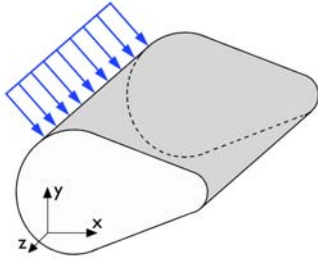


Figure 2-3: A geometry suitable for plane strain analysis.

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  $y$ . The plane strain condition prevails in geometries, whose extent is large in the  $z$  direction compared to in the  $x$  and  $y$  directions, or when the  $z$  displacement is in some way restricted. One example is a long tunnel along the  $z$ -axis where it is sufficient to study a unit-depth slice in the  $xy$ -plane.

### AXISYMMETRIC GEOMETRY

The axisymmetric variant of the Solid Mechanics interface uses cylindrical coordinates  $r$ ,  $\phi$  ( $\phi$ i), and  $z$ . Loads are independent of  $\phi$ , and the axisymmetric variant of the physics interface allows loads only in the  $r$  and  $z$  directions.

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

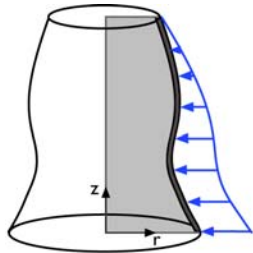


Figure 2-4: Rotating a 2D geometry to recover a 3D solid.

## *Elastic Waves, Time Explicit*

---

The Elastic Waves, Time Explicit 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 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 it can 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 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 stresses and 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.



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. The material model is linear elastic.

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 a 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. It is then better to use the Membrane interface.

### *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.

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 only exception is when it is used as a “cladding” on top of a solid, since it will then be stabilized by the solid.

In the Membrane interface a large number of different material models can be used.

### *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 specified in terms of properties such as area and moments of inertia.

The exact stress distribution in the beam is not explicitly modeled. It is actually not even fully determined by 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.

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.

## *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 interface has four distinct purposes:

- Modeling of trusses, consisting of straight bars carrying only axial forces
- Modeling of cables and wires
- As reinforcements, used in conjunction with other physics interfaces
- For modeling discrete springs and dampers, often added to other physics interfaces

For a truss model, only one geometrical property is needed, the cross-section area. The material model can be linear elastic or elastoplastic. There is also a special material model for creating spring/damper data.

The 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 cables however, a prestress is necessary to keep them stable.

# 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 tradeoff 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.

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.

If the solution contains discontinuities, like 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 be 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 to other 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:

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

Since the thermal strain is directly proportional to the temperature, a consistent approximation would be to use one order lower discretization order for the temperature than for the displacements. These kinds of considerations are automatically made when you add the built-in multiphysics interfaces, such as [The Thermal Stress, Solid Interface](#).

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.

### *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.

#### **MIXED FORMULATION**

When the mix formulation is selected for a certain material, the pressure (mean stress) is added as an extra degree of freedom to form what is called a *mixed formulation*.

The shape function used for the pressure is continuous Lagrangian, having an order that is one below what is used for the displacement shape functions. The pressure degree of freedom is, however, discontinuous over domain boundaries in order to allow for discontinuities in material properties and stresses.

#### **VISCOELASTIC STRESS AND STRAIN**

When using the Viscoelastic material, 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.

### **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 as used for the numerical integration of the stiffness matrix. This order depends on the selected displacement discretization order.

# Coupling Different Element Types

In this section:

- [Introduction to the Element Types](#)
- [Coupling Techniques](#)

## *Introduction to the Element Types*

---

In some engineering 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 ( $u_2$ ,  $v_2$ ,  $w_2$ ), 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 physics interfaces. This is sufficient, for example, when using membranes as cladding on a solid boundary or truss elements as reinforcement bars in a solid.

In the Beam, Shell, and Plate interfaces, the deformation is described also by 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 next.



---

The default shape functions in the Solid Mechanics interface are of the serendipity type, whereas in the Shell interface Lagrange shape functions are used. 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 interface have special properties, and a beam cannot have the same degrees of freedom as another physics interface if the same edge or boundary is shared.

Also, the representation of rotations differs between the Shell and Plate interfaces (displacement of normal) and the Beam interface (rotation angle). It is therefore not possible to use common degree of freedom names for 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 difficult to set up manually:

- [Shell Edge to Solid Boundary \(3D\)](#)
- [Shell Boundary to Solid Boundary \(3D\)](#)
- [Membrane Boundary to Solid Boundary \(3D\)](#)



- 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)*

A shell can be coupled to 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)*

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)*

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, deselect 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 significantly 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. This you can do 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** check box.

#### *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, beam, and membrane elements to a Solid Mechanics interface. 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](#).

---

	<p>The underlying theory and more details about the built-in couplings can be found in</p> <ul style="list-style-type: none"><li>• <a href="#">Connection Between Shells and Solids</a></li><li>• <a href="#">Connection Between Shells and Beams</a></li><li>• <a href="#">Embedded Elements</a></li></ul>
	<ul style="list-style-type: none"><li>• Examples of all types of couplings between shells and beams are shown in <i>Connecting Shells and Beams</i>: Application Library path <b>Structural_Mechanics_Module/Beams_and_Shells/shell_beam_connection</b></li><li>• An example of couplings between shells and solids is shown in <i>Connecting Shells and Solids</i>: Application Library path <b>Structural_Mechanics_Module/Beams_and_Shells/shell_solid_connection</b></li><li>• An example of couplings between beams and solids is shown in <i>Connecting Beams and Solids</i>: Application Library path <b>Structural_Mechanics_Module/Beams_and_Shells/beam_solid_connection</b></li></ul>

---

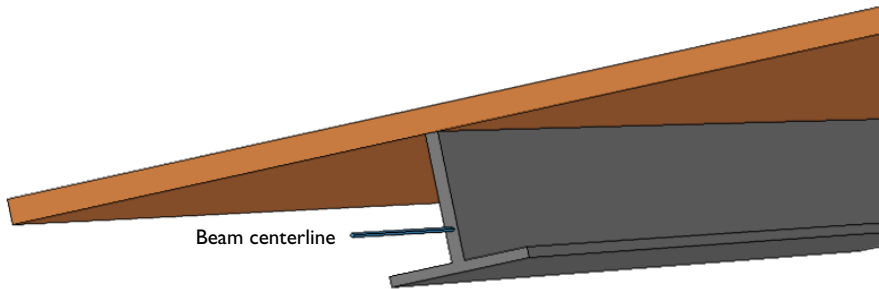
### USING GENERAL COUPLING OPERATORS

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 displays 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,  $\phi$  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  $\mathbf{n}$ .

To create the coupling:

- I Add a **General Extrusion** node under **Definitions**. 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  $\text{genext1}(u) + \text{genext1}(\text{shell.thy}) * \text{zdist}$ , where  $\text{zdist}$  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](#)
-

# 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 domains or rigid connectors.

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](#)

## *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 in random vibration analysis. In that case, no automatic unit conversions are available.

### **PREDEFINED AND CUSTOM COORDINATE SYSTEMS**

In this module, 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 and 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** check box is selected under the **Study settings** section of the current study step.

### 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 check box **Show physics symbols**, by which you can control the display of the symbols for that specific feature.

Loads are among the results for which default plots 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 default plots. The fastest way to do that is

to click **Get Initial Value** on 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 study, you can define load cases and constraint cases. A load or constraint can be assigned to a load or constraint group, and then 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 off the whole constraint. The load group acts as a multiplier to nonzero prescribed values of displacement and rotations.
- 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.
- Since **Thermal Expansion** nodes are exclusive (only the last one given gives a contribution for a certain domain), you cannot switch between different **Thermal Expansion** nodes only by assigning them to different load cases.
- **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 it 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 Domain**, 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 is if you want to avoid that `expr` is evaluated.

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 *Pratt Truss Bridge*: Application Library path **Structural\_Mechanics\_Module/Beams\_and\_Shells/pratt\_truss\_bridge**.

---

### *Singular Loads*

In reality, loads always act on a finite area. However, in a model a load is sometimes defined on a point or an edge, 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 stresses depend on both the mesh and the material properties. The stress distribution at locations far from these singularities is unaffected according to a well-known principle in solid mechanics, the St. 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 2-5](#) 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.



The **Plasticity** and **Creep** nodes are available as a subnode to **Linear Elastic Material** nodes with the Nonlinear Structural Materials Module or the Geomechanics Module.

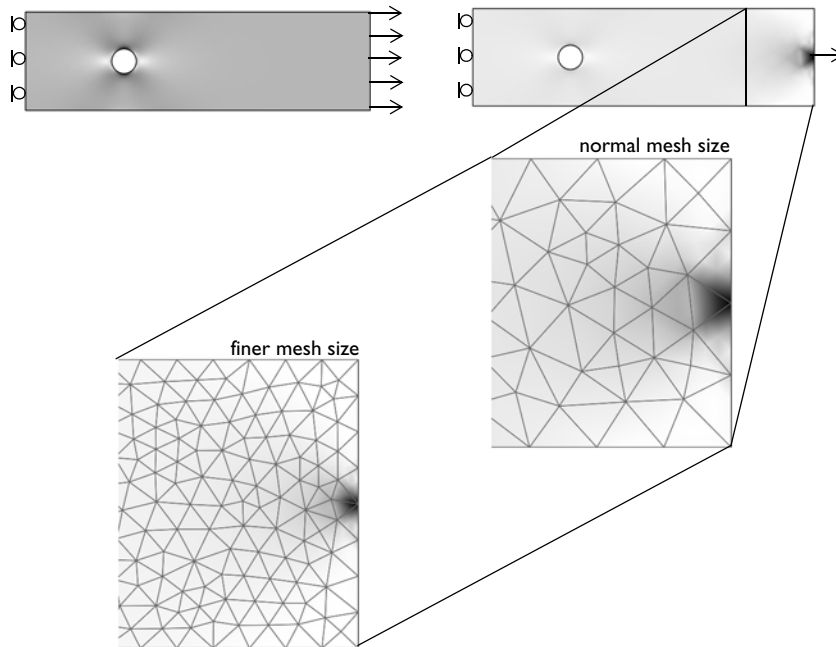


Figure 2-5: A plate with a hole subject to a distributed load (left) and a point load (right).

### *Moments in the Solid Mechanics Interface*

The Solid Mechanics interface, as opposed to the Beam, 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** check box 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 loads like **Gravity** and **Rotating Frame**. 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. Such loads are however not applied to mass contributions that belong to global features such as rigid connectors.

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 a an acceleration force, using a **Gravity** node. In particular, this is necessary when using modal based dynamic analyses, in which non-zero prescribed displacements cannot be used.

### *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 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 it 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 VARIATION

If a load has a 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 if you increase the integration order.

## 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.bnd11.F_Ax`.

TABLE 2-4: THE LOAD VARIABLE NAMING SCHEME

	DESCRIPTION	EXAMPLES
<code>&lt;phys&gt;</code>	Physics interface tag	solid, truss2
<code>&lt;loadTag&gt;</code>	Tag of the load feature	bnd11, e12
<code>&lt;loadType&gt;</code>	Force or moment	F (force) M (moment)



TABLE 2-4: THE LOAD VARIABLE NAMING SCHEME

	DESCRIPTION	EXAMPLES
<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 do not have a geometrical selection, such as loads on rigid connectors and rigid domains, 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. This module has 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](#)

## *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.

The number of possible rigid body modes for different geometrical dimensions is shown in the table below.

TABLE 2-5: 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)

If the model is underconstrained, 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](#) boundary condition 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 Boundary Condition](#) 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  $\omega$  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  $90^\circ$ . 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

$$u_p(t) = u_0(t_0) + \int_{t_0}^t v_p(\tau) d\tau$$

or

$$u_p(t) = u_0(t_0) + \int_{t_0}^t \left( v_0(t_0) + \int_{t_0}^{\tau} a_p(\zeta) d\zeta \right) d\tau$$

where  $u_0$  and  $v_0$  are 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 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 2-6), model a quarter of the geometry by using the *Symmetry* node for the two selected surfaces.

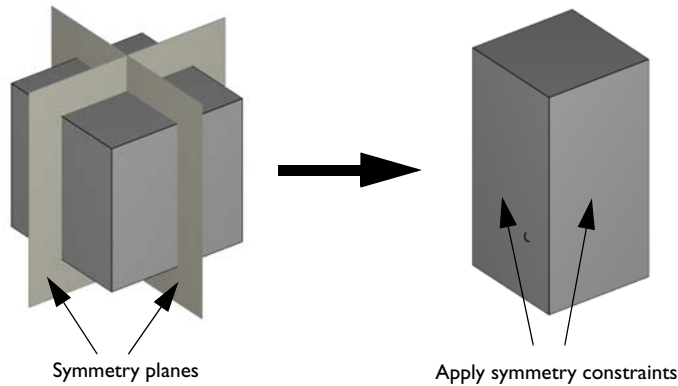


Figure 2-6: If the geometry exhibits two symmetry planes, model a quarter of the geometry by using the *Symmetry* feature for the two selected surfaces.



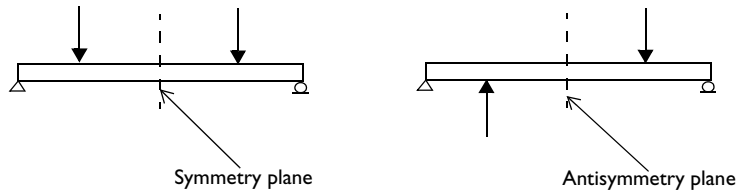
Both geometric symmetry and loads are important when selecting the correct constraints for a model.



In an eigenfrequency or buckling analysis, the eigenmodes might be nonsymmetric even if the structure is symmetric.

Figure 2-7 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 2-7: 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 be 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 *Thermo-Mechanical 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 2-6](#), 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*

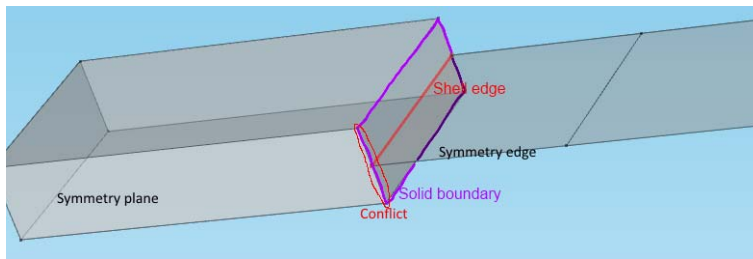
---

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 2-8](#).



*Figure 2-8: 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 in 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 node 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.

# 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](#)
- [Using Surface Traction to Evaluate Reaction Forces](#)

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**. 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).

Specify the default coordinates of the **Reference Point for Moment Computation** at the top level of the **Settings** window for the physics interface. After editing the reference point coordinates, you need to right-click the Study node and select **Update Solution** for the change to take effect on the reaction moment calculation. During

postprocessing, 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.
- 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*

Select the **Use weak constraints** check box to get 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  $x_{1m}$ , where  $x$  is the name of the constrained degree of freedom (as, for example,  $u_{1m}$  and  $v_{1m}$ ). The extension  $1m$  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  $x_{1m}$  using **Volume Integration**, **Surface Integration**, or **Line Integration** under **Derived Values**.

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*:

- [Derived Values, Evaluation Groups, and Tables](#)
- [Symmetric and Nonsymmetric Constraints](#)

### *Using Surface Traction to Evaluate Reaction Forces*

As an alternative method, 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 components, multiply the surface traction by the cross-section thickness 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  $interface.Tax$ , 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  $interface.Tracx$ , is computed using a method similar to the weak constraints, but without introducing the Lagrange multipliers as extra degrees of freedom. The accuracy is high, but there is an extra computational cost. These traction variables are computed only if the **Compute boundary fluxes** check box 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  $interface.Tax$  variables are based on Cauchy stress, and contains a force per current area. If you integrate them you must use the spatial frame. The  $interface.Tracx$  variables are based on First Piola-Kirchhoff stresses and contains a force per undeformed area. An integration must then be done on the material frame.

### *Evaluating Surface Traction Forces on Interior Boundaries*

As opposed to the other methods for reaction force computation, the boundary flux based tractions are computed not only on external boundaries, but also on interior boundaries. On interior boundaries, there are then two traction fields: One acting from each of the domains sharing the boundary. These internal traction fields are contained in the variables  $interface.iTracux$  and  $interface.iTracdx$ . 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. The  $interface.iTracux$  and  $interface.iTracdx$  variables are only available if the **Compute boundary fluxes** check box is selected in the **Discretization** section of the physics interface, and there are interior boundaries in your model.



[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 [Table 2-6](#) and [Table 2-7](#) you can find an overview of the families of materials, and their applicability in the various structural mechanics interfaces.

TABLE 2-6: MATERIAL MODELS IN THE PHYSICS INTERFACES ON DOMAINS AND BOUNDARIES

MATERIAL MODEL	SOLID MECHANICS	ELASTIC WAVES, TIME EXPLICIT	SHELL/ PLATE	LAYERED SHELL	MEMBRANE
Linear Elastic Material	X	—	X	X	X
Layered Linear Elastic Material	—	—	X	—	X
Section Stiffness	—	—	X	—	—
Elastic Waves, Time Explicit Model	—	X	—	—	—
Nonlinear Elastic Material	X	—	—	—	X
Hyperelastic Material	X	—	—	X	X
Shape Memory Alloy	X	—	—	—	—
Piezoelectric Material	X	—	—	X	—
Magnetostrictive Material	X	—	—	—	—
Viscoelasticity	X	—	X	X	X
Plasticity	X	—	X	X	X
Soil Plasticity	X	—	—	—	—
Creep	X	—	X	X	X
Elastoplastic Soil Material	X	—	—	—	—
Viscoplasticity	X	—	X	X	X
Porous Plasticity	X	—	—	—	—
Concrete	X	—	—	—	—
Rocks	X	—	—	—	—
Damage	X	—	—	—	—
External Stress-Strain Relation	X	—	—	—	X
Rigid Domain	X	—	X	X	—

TABLE 2-7: MATERIAL MODELS IN THE PHYSICS INTERFACES ON EDGES

MATERIAL MODEL	BEAM	PIPE MECHANICS	TRUSS
Linear Elastic Material	X	—	X
Fluid and Pipe Materials	—	X	—
Plasticity	—	—	X
Rigid Domain	X	X	—
Spring-Damper Material	—	—	X

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
- Two different creep models
- 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  $\nu_{xy}$  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 Viscoelastic Materials](#)
  - [Viscoelasticity](#)
-

## Mixed Formulation

---

Nearly 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.



The mixed formulation is useful not only for linear elastic materials but also for elastoplastic materials, hyperelastic materials, and viscoelastic materials.

---



The order of the shape function for the auxiliary pressure variable should be one order less than that for the displacements. Thus, it is not recommended to use linear elements for the displacement variables in the domains, where the mixed formulation is turned on. Also 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 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 or in the Nonlinear Structural Mechanics 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 or in the Nonlinear Structural Mechanics 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 unit area, and  $\sigma_{ts}$  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.

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  $n+1$ :

- I *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.

An improvement to the method is to add a multi-pass 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 or in the Nonlinear Structural Mechanics Module Application Libraries, where a multi-pass 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 sub group 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 Materials 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 be 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 un-altered 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);
```

# Modeling Piezoelectric Problems

In this section:

- [About Piezoelectric Materials](#)
- [Piezoelectric Coupling](#)
- [Create the Piezoelectric Effect 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](#)
- [Piezoelectric Material](#) 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:

$$\begin{aligned} \mathbf{S} &= s_E \mathbf{T} + d^T \mathbf{E} \\ \mathbf{D} &= d \mathbf{T} + \epsilon_T \mathbf{E} \end{aligned} \quad (2-16)$$

Here, the naming convention used in piezoelectricity theory is assumed:  $\mathbf{S}$  is the strain,  $\mathbf{T}$  is the stress,  $\mathbf{E}$  is the electric field, and  $\mathbf{D}$  is the electric displacement field. The material parameters  $s_E$ ,  $d$ , and  $\epsilon_T$ , 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  $\varepsilon$  (instead of  $\mathbf{S}$ ) and the stresses are denoted by either  $\sigma$  or  $S$  (instead of  $\mathbf{T}$ ). This makes the names consistent with those used in the other structural mechanics interfaces.

Equation 2-16 will, using the notation from structural mechanics, then read

$$\begin{aligned}\varepsilon &= s_E \sigma + d^T \mathbf{E} \\ \mathbf{D} &= d \sigma + \varepsilon_0 \varepsilon_{rT} \mathbf{E}\end{aligned}\tag{2-17}$$

Equation 2-16 (or Equation 2-17) is known as the *strain-charge form* of the constitutive relations. The equation can be re-arranged into the *stress-charge form*, which relates the material stresses to the electric field:

$$\begin{aligned}\sigma &= c_E \varepsilon - e^T \mathbf{E} \\ \mathbf{D} &= e \varepsilon + \varepsilon_0 \varepsilon_{rS} \mathbf{E}\end{aligned}\tag{2-18}$$

The material properties,  $c_E$ ,  $e$ , and  $\varepsilon_S$  are related to  $s_E$ ,  $d$ , and  $\varepsilon_T$ . It is possible to use either form of the constitutive relations. In addition to Equation 2-16 or Equation 2-18, the equations of solid mechanics and electrostatics must also be solved within the material.



- [Piezoelectric Coupling](#)
- [Modeling Piezoelectric Problems](#)
- [Piezoelectric Material](#) 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 2-8). 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:  $(YXl) - 35.25^\circ$ . 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  $Z$  axis, is required. This is illustrated in [Figure 2-10](#). Note that within the 1949 IRE Standard AT cut quartz is denoted as:  $(YXl) +35.25^\circ$ , since the  $X$ -axis is rotated by  $180^\circ$  in this convention and positive angles therefore correspond to the opposite direction of rotation (see [Figure 2-9](#)). [Table 2-9](#) 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 2-9](#) 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 2-8](#) summarizes the signs for the important matrix elements under the two conventions. [Table 2-9](#) shows the different definitions of the crystal cuts under the two conventions.

TABLE 2-8: SIGNS FOR THE MATERIAL PROPERTIES OF QUARTZ, WITHIN THE TWO STANDARDS COMMONLY EMPLOYED.

MATERIAL PROPERTY	IRE 1949 STANDARD		IEEE 1978 STANDARD	
	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 2-9: 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	$(YXl) +35.25^\circ$	$(YXl) -49^\circ$
	Y-thickness Euler angles	$(ZXZ: 0^\circ, -35.25^\circ, 0^\circ)$	$(ZXZ: 0^\circ, 49^\circ, 0^\circ)$
	Z-thickness Euler angles	$(ZXZ: 0^\circ, -125.25^\circ, 0^\circ)$	$(ZXZ: 0^\circ, -41^\circ, 0^\circ)$
IEEE 1978	Standard	$(YXl) -35.25^\circ$	$(YXl) +49^\circ$



TABLE 2-9: 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	( $ZXZ: 0^\circ, 35.25^\circ, 0^\circ$ )	( $ZXZ: 0^\circ, -49^\circ, 0^\circ$ )
	Z-thickness Euler angles	( $ZXZ: 0^\circ, -54.75^\circ, 0^\circ$ )	( $ZXZ: 0^\circ, -139^\circ, 0^\circ$ )

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 2-10](#). 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 orientate its normal parallel to the  $Y$  axis in the global coordinate system. [Figure 2-11](#) shows how to define the local coordinate system in this case. [Figure 2-12](#) 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.

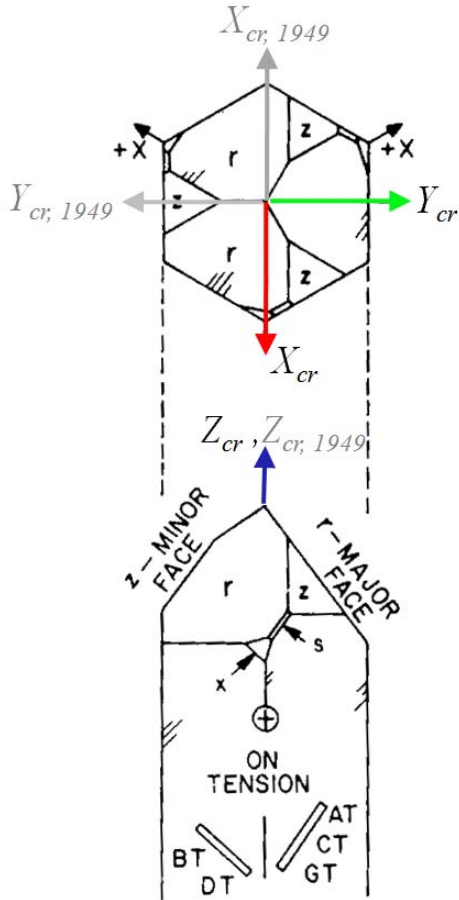


Figure 2-9: Crystallographic axes defined for right-handed quartz in COMSOL and the 1978 IEEE standard (color). The 1949 standard axes are shown for comparison (gray).



Figure 2-9 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.

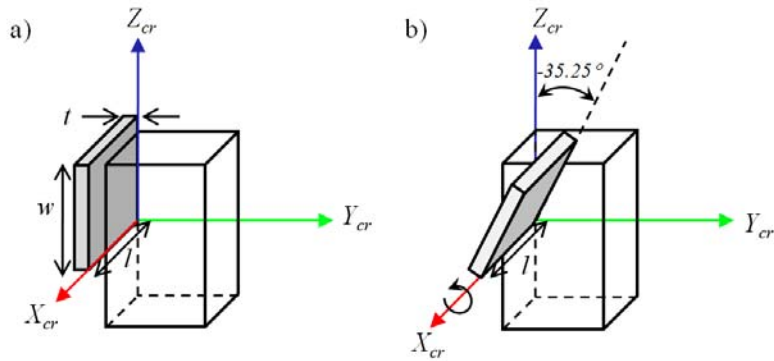


Figure 2-10: Definition of the AT cut of quartz within the IEEE 1978 standard. The AT cut is defined as:  $(YXL) -35.25^\circ$ . The first two bracketed letters specify the initial orientation of the plate, with the thickness direction,  $t$ , along the crystal  $Y$  axis and the length direction,  $l$ , 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  $-35.25^\circ$  (in a right-handed sense).

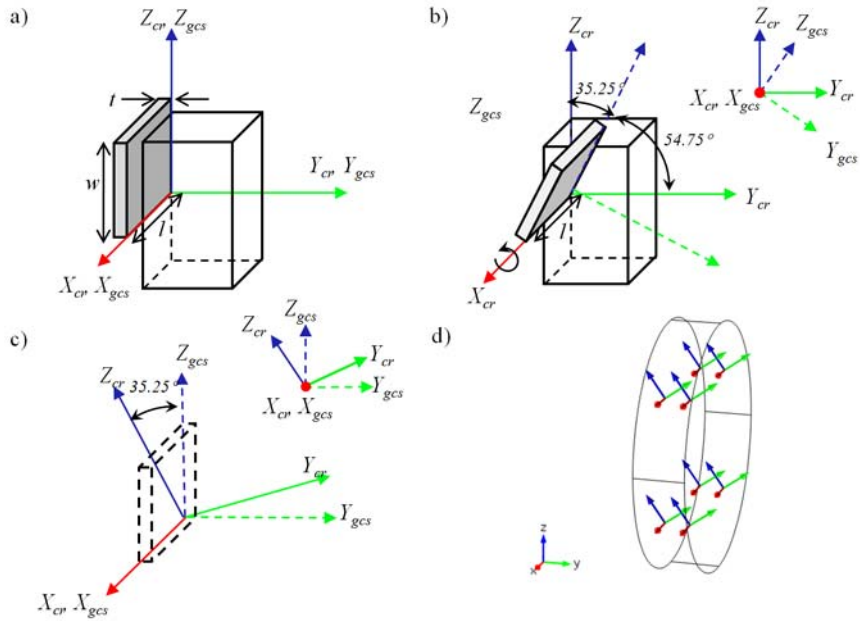


Figure 2-11: Defining an AT cut crystal plate within COMSOL, with normal in the global  $Y$  direction. Within the 1978 IEEE standard the AT cut is defined as  $(YXU) -35.25^\circ$ . Start with the plate normal or thickness in the  $Y_{cr}$  direction (a) and rotate the plate  $-35.25^\circ$  about the  $l$  axis (b). The global coordinate system rotates with the plate. Finally rotate the entire system so that the global coordinate system is orientated as it appears in COMSOL (c). The local coordinate system should be defined with the Euler angles  $(ZXZ - 0, 35.25^\circ, 0)$ . (d) shows a coordinate system for this system in COMSOL.

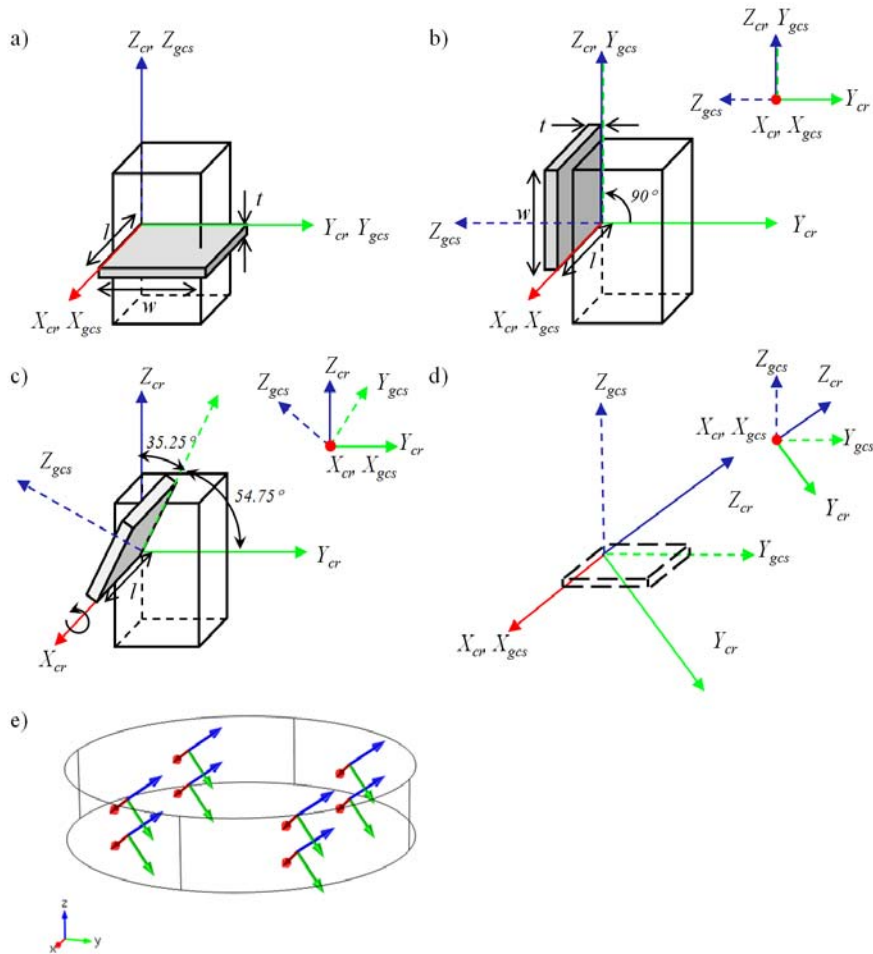


Figure 2-12: Defining an AT cut crystal plate within COMSOL, with normal in the global  $Z$  direction. Within the 1978 IEEE standard the AT cut is defined as (YXL)  $-35.25^\circ$ . Begin with the plate normal in the  $Z_{cr}$  direction, so the crystal and global systems are coincident. Rotate the 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  $-35.25^\circ$  about the  $l$  axis (d). Finally rotate the entire system so that the global coordinate system is orientated as it appears in COMSOL (d). The local coordinate system should be defined with the Euler angles (ZXZ:  $0, -54.75^\circ, 0$ ). (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 [Electrical 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, and by using the **Dielectric Dispersion** option in the **Dielectric Loss** subnodes. 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 the case of piezoelectric materials, this means that the constitutive equations are written as follows.

For the stress-charge formulation

$$\begin{aligned}\boldsymbol{\sigma} &= \tilde{c}_E \boldsymbol{\varepsilon} - \tilde{e}^T \mathbf{E} \\ \mathbf{D} &= \tilde{e} \boldsymbol{\varepsilon} + \tilde{\varepsilon}_S \mathbf{E}\end{aligned}$$

and for the strain-charge formulation

$$\begin{aligned}\varepsilon &= \tilde{s}_E \sigma + \tilde{d}^T \mathbf{E} \\ \mathbf{D} &= \tilde{d} \sigma + \tilde{\varepsilon}_T \mathbf{E}\end{aligned}$$

where  $\tilde{c}_E$ ,  $\tilde{d}$ , and  $\varepsilon$  are complex-valued matrices, where the imaginary part defines 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

$$X = \text{real}(\tilde{X})$$

and a loss factor

$$\eta_X = \text{imag}(\tilde{X})/\text{real}(\tilde{X})$$

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,  $\eta_{cE}$ . For a structural material without coupling, simply use  $\eta_s$ , 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*

$$\tilde{\epsilon}_{rS}^{m,n} = \epsilon_{rS}^{m,n} (1 - j\eta_{\epsilon S}^{m,n})$$

$$\tilde{\epsilon}_{rT}^{m,n} = \epsilon_{rT}^{m,n} (1 - j\eta_{\epsilon T}^{m,n})$$

Note that the multiplication is applied component-wise.

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.

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.

For more information about hysteretic losses, see [Ref. 1](#) to [Ref. 4](#).



For more details, see [Piezoelectric Dissipation](#) in the Structural Mechanics Theory chapter.

---



### ELECTRICAL CONDUCTIVITY (TIME HARMONIC)

For frequency domain and eigenfrequency analyses, the effect of electrical 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  $\sigma_e$  is the material electrical 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. However, the dielectric dispersion can be used together with conduction loss in an eigenfrequency analysis, as shown in the next section [Dielectric Dispersion](#).



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 electrical 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](#)).

## DIELECTRIC DISPERSION

The **Dielectric Loss** subnode can be set to use the **Dielectric Dispersion** option. In such case, the following equations need to be solved in the time domain:

$$\nabla \cdot \left( \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}_p \right) = 0$$

$$\mathbf{J}_p + \tau_d \frac{\partial \mathbf{J}_p}{\partial t} = \epsilon_0 \Delta \epsilon_{rS} \frac{\partial \mathbf{E}}{\partial t}$$

where you can specify two material parameters: the relaxation time  $\tau_d$  and the relative permittivity increment  $\Delta \epsilon_{rS}$ . 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](#).

With the absence of free electric charges, the above equations can be combined and integrated in time to yield the following equation:

$$\nabla \cdot \left( \mathbf{D} + \tau_d \frac{\partial \mathbf{D}}{\partial t} + \epsilon_0 \Delta \epsilon_{rS} \mathbf{E} \right) = 0$$

This is the equation form used in COMSOL Multiphysics for time dependent analysis.

For the eigenfrequency and frequency domain analyses, the corresponding equation is:

$$\nabla \cdot \left[ (1 + j\tau_d \omega) j\omega \mathbf{D} + \epsilon_0 \Delta \epsilon_{rS} j\omega \mathbf{E} \right] = 0$$

In most cases, the factor  $j\omega$  can be factored out, so that the following equation is solved:

$$\nabla \cdot \left( \mathbf{D} + j\tau_d \omega \mathbf{D} + \epsilon_0 \Delta \epsilon_{rS} \mathbf{E} \right) = 0$$

This equation, together with the constitutive relation  $\mathbf{D} = e\epsilon + \epsilon_0 \epsilon_{rS} \mathbf{E}$ , gives

$$\nabla \cdot \left[ e\epsilon + \epsilon_0 \left( \epsilon_{rS} + \frac{\Delta \epsilon_{rS}}{1 + (\tau_d \omega)^2} \right) \mathbf{E} - j \frac{\tau_d \omega \epsilon_0 \Delta \epsilon_{rS}}{1 + (\tau_d \omega)^2} \mathbf{E} \right] = 0$$

which shows how the dispersion parameters contribute to the polarization and losses. Thus, the effective relative permittivity varies from  $\epsilon_{rS} + \Delta \epsilon_{rS}$  down to  $\epsilon_{rS}$  as the excitation frequency increases from zero. The damping effect vanishes for both large and small frequencies, and it reaches the maximum for  $\omega = 1/\tau_d$ .

Dielectric dispersion can be combined with conduction loss for both eigenfrequency and frequency domain analyses. The following equation forms are used, respectively, in the frequency domain:

$$\nabla \cdot [\mathbf{D} + j\tau_d\omega\mathbf{D} + \varepsilon_0\Delta\varepsilon_{rS}\mathbf{E} + (\tau_d - j\omega^{-1})\mathbf{J}_c] = 0$$

and in eigenfrequency analyses:

$$\nabla \cdot [(1 + j\tau_d\omega)j\omega\mathbf{D} + \varepsilon_0\Delta\varepsilon_{rS}j\omega\mathbf{E} + (1 + j\tau_d\omega)\mathbf{J}_c] = 0$$

where  $\mathbf{J}_c = \sigma_c\mathbf{E}$  is the conduction electric current.

### *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 **Piezoelectric Effect** 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 Piezoelectric Effect 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>Piezoelectric Effect** 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 **Piezoelectric Effect** 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. On 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 check box to force strains to be linear.
- **Energy Dissipation**: Select the check box 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  $c\mathbf{E}$ , a loss factor for  $s\mathbf{E}$  (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  $\epsilon\mathbf{S}$ , a loss factor for  $\epsilon\mathbf{T}$  (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 Electrical 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:  $\text{kg}/\text{m}^3$ )
- Elasticity matrix  $c\mathbf{E}$  (SI unit: Pa) in Stress-charge form.
- Compliance Matrix  $s\mathbf{E}$  (SI unit:  $1/\text{Pa}$ ) in Strain-charge form.

### **ELECTROSTATIC PROPERTIES**

- Relative Permittivity  $\epsilon_{\text{S}}$  (dimensionless) in Stress-charge form.
- Relative Permittivity  $\epsilon_{\text{T}}$  (dimensionless) in Strain-charge form.

### **COUPLING PROPERTIES**

- Coupling matrix  $e\mathbf{ES}$  (SI unit:  $\text{C}/\text{m}^2$ ) in Stress-Charge form.
- Coupling matrix  $d\mathbf{ET}$  (SI unit:  $\text{C}/\text{N}$ ) in Strain-charge form.

## DAMPING AND LOSS PROPERTIES

- Loss factor for elasticity matrix  $cE$ ,  $\eta_{cE}$ , or loss factor for elasticity matrix  $sE$ ,  $\eta_{sE}$ : required when Mechanical Damping is present. The latter is valid only in Strain-Charge form.
- Loss factor for electrical permittivity matrix  $\epsilon S$ ,  $\eta_{\epsilon S}$ , or Loss factor for electrical permittivity matrix  $\epsilon T$ ,  $\eta_{\epsilon T}$ : required when Dielectric Loss is present. The latter is valid only in Strain-Charge form.
- Loss factor for coupling matrix  $e$ ,  $\eta_{eS}$ , or loss factor for coupling matrix  $d$ ,  $\eta_{dT}$ : required when Coupling Loss is present. The latter is valid only in Strain-Charge form.
- Electrical conductivity sigma (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 **Piezoelectric Effect** 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), and a stress applied on it changes the material magnetization (inverse effect). In COMSOL Multiphysics, the Magnetostriction interface is constituted of one **Solid Mechanics** and one **Magnetic Fields** interface, which are coupled together via a **Magnetostriction** multiphysics 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.

## *Create the Magnetostriction Interface and Define Domains*

---

A magnetostriction problem can be set up in different ways:

- By selecting **Magnetostriction** from the **Model Wizard**,
- By choosing **Magnetostriction** 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 magnetostrictive materials.

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.

- 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.
- 2 Go to the **Solid Mechanics>Magnetostrictive Material** node (if it is not yet available, add it). Select the domains where the magnetostriction effect applies. Nonmagnetostrictive domains can be modeled using the other available material models.

- 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 magnetostrictive domains, and any other magnetic domains.
- 4 Go to the **Magnetic Fields>Ampere's Law, Magnetostrictive** node (if it is not yet available, add it). Select the domains where the magnetostriction effect has to be modeled.
- 5 A **Multiphysics>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 **Magnetostriction** coupling.
- 6 Confirm that all the domains where the magnetostrictive material is present are selected.



---

Only domains that have both **Ampere's Law, Magnetostrictive** selected in the **Magnetic Fields** interface and **Magnetostrictive Material** selected in the **Solid Mechanics** interface are active in the selection for **Magnetostriction** coupling. The selection of this multiphysics feature cannot be edited. If several **Solid Mechanics** or **Magnetic Fields** interfaces are present, select the correct ones.

---

- 7 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 the magnetostriction coupling is negligible.
  - Solid-only domains: In these domains, only mechanical phenomena are modeled, and the magnetics phenomena are neglected. This represents deformable solid nonmagnetic materials.
  - Magnetics-only domains: Those domains are nonsolid. This is typically the case of air, in which the magnetics is solved but not mechanics.
  - 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 various multiphysics models.

## *Linear vs. Nonlinear Magnetostriction*

---

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).

For linear magnetostriction model, it is possible to express the relation between the stress  $\mathbf{S}$ , strain  $\epsilon$ , 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 all necessary transformations if needed. For details, see the corresponding theory section [Linear Magnetostriction Model](#). You find all the necessary material data inputs within the **Magnetostrictive 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. For more details, see the corresponding theory section [Nonlinear Magnetostriction](#).

In 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.

# 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:

$$m \frac{d^2 u}{dt^2} + c \frac{du}{dt} + ku = f(t) \quad (2-19)$$

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:

$$\frac{d^2 u}{dt^2} + 2\zeta\omega_0 \frac{du}{dt} + \omega_0^2 u = \frac{f(t)}{m} \quad (2-20)$$

where  $\zeta = c/(2m\omega_0)$  and  $\omega_0^2 = k/m$ . In this case  $\zeta$  is the *damping ratio* ( $\zeta = 1$  for critical damping) and  $\omega_0$  is the *undamped resonant frequency* of the system. In the literature it is more common to give values of  $\zeta$  than  $c$ . The damping ratio  $\zeta$  can also be readily related to many of the various measures of damping employed in different disciplines. These are summarized in [Table 2-10](#).

TABLE 2-10: RELATIONSHIPS BETWEEN MEASURES OF DAMPING

DAMPING PARAMETER	DEFINITION	RELATION TO DAMPING RATIO
Damping ratio	$\zeta = c/c_{\text{critical}}$	–
Logarithmic decrement	$\delta_d = \ln\left(\frac{u(t_0)}{u(t_0 + \tau)}\right)$ <p>where <math>t_0</math> is a reference time and <math>\tau</math> is the period of vibration for a decaying, unforced degree of freedom.</p>	$\delta_d \approx 2\pi\zeta$ <p>(<math>\zeta \ll 1</math>)</p>



TABLE 2-10: 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 <code>solid.Qh</code> and <code>solid.Wh</code> .	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

$$f(t) = \text{Re}\{F e^{j\omega t}\} \text{ and } u(t) = \text{Re}\{U e^{j\omega t}\}$$

are written where  $\omega$  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 2-19 takes the following form in the frequency domain:

$$-\omega^2 m U + j\omega c U + k U = F \tag{2-21}$$

where the time dependence has canceled out on both sides. Alternatively, this equation can be written as:

$$-\omega^2 U + 2j\omega\omega_0\zeta U + \omega_0^2 U = \frac{F}{m} \tag{2-22}$$

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 2-19](#). This means that the method can be applied generally in either the time or frequency domain. The parameter  $c$  in [Equation 2-19](#) is defined as a fraction of the mass and the stiffness using two parameters,  $\alpha_{dM}$  and  $\beta_{dK}$ , such that

$$c = \alpha_{dM}m + \beta_{dK}k \quad (2-23)$$

Substituting this relationship into [Equation 2-19](#) and rearranging into the form of [Equation 2-20](#) gives:

$$\frac{d^2u}{dt^2} + (\alpha_{dM} + \beta_{dK}\omega_0^2)\frac{du}{dt} + \omega_0^2u = \frac{f(t)}{m}$$

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:

$$\zeta = \frac{1}{2}\left(\frac{\alpha_{dM}}{\omega_0} + \beta_{dK}\omega_0\right) \quad (2-24)$$

Note that [Equation 2-24](#) 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  $\alpha_{dM}$  and  $\beta_{dK}$ . For example, setting  $\alpha_{dM} = 0$  and  $\beta_{dK} = 2\zeta/\omega_0$  produces an equivalent viscous damping model at the resonant frequency  $\omega_0$ .

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 2-21](#)). 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  $\alpha_{dM}$  and  $\beta_{dK}$ . 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 2-24, this relationship at two frequencies,  $f_1$  and  $f_2$ , with different damping ratio,  $\zeta_1$  and  $\zeta_2$ , results in an equation system that can be solved for  $\alpha_{dM}$  and  $\beta_{dK}$ :

$$\begin{bmatrix} \frac{1}{4\pi f_1} & \pi f_1 \\ \frac{1}{4\pi f_2} & \pi f_2 \end{bmatrix} \begin{bmatrix} \alpha_{dM} \\ \beta_{dK} \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}$$

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 2-13 shows.

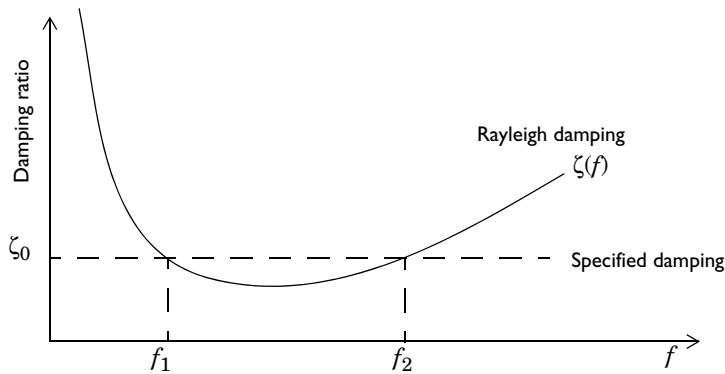


Figure 2-13: An example of Rayleigh damping.

Since the coefficients  $\alpha_{dM}$  and  $\beta_{dK}$  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  $\alpha_{dM}$  as zero and to define damping only using the  $\beta_{dK}$  coefficient. Then, according to Equation 2-24, a damping which 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  $\beta_{dK}$  is:

$$\beta_{dK} = \zeta / (\pi f_0) = \eta / (2\pi f_0)$$

### *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 unit volume.

In COMSOL Multiphysics the loss information appears as a multiplier to the elastic constitutive matrix  $D^c$

$$D^c = (1 + j\eta_s)D$$

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  $\eta_s$ . But there is no reason that  $\eta_s$  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.

$$D_{mn}^c = (1 + j\eta_{s, mn})D_{mn}$$

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, an extra contribution, proportional to the strain rate, is added to the stress tensor, as described in [Viscous Damping Theory](#).

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  $\beta_{dK}$ , to the loss factor,  $\eta$ , divided by the excitation frequency:

$$\beta_{dK} = \frac{\eta}{2\pi f} = \frac{\eta}{\omega}$$

The mass damping factor,  $\alpha_{dM}$ , 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

$$\begin{aligned}\eta_b &= K\beta_{dK} \\ \eta_v &= G\beta_{dK}\end{aligned}$$

where  $K$  and  $G$  are the bulk and shear moduli, respectively. Equivalently, you can transform between loss factor damping and viscous damping,

$$\begin{aligned}\eta_b &= \frac{K\eta}{\omega} \\ \eta_v &= \frac{G\eta}{\omega}\end{aligned}$$

### *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.

### *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  $\eta_s$  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,  $\beta_{dK}$  is only used as a multiplier of the structural contribution to the stiffness matrix when building-up the damping matrix as given by [Equation 2-23](#). 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 2-14](#)). In this case the deformation of the beam introduces an effect known as *geometric nonlinearity* into the equations of solid mechanics.

[Figure 2-14](#) 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 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 components  $u$ ,  $v$ , and  $w$ . At a given coordinate  $(X, Y, Z)$  in the reference configuration (on the left of [Figure 2-14](#)) 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 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:

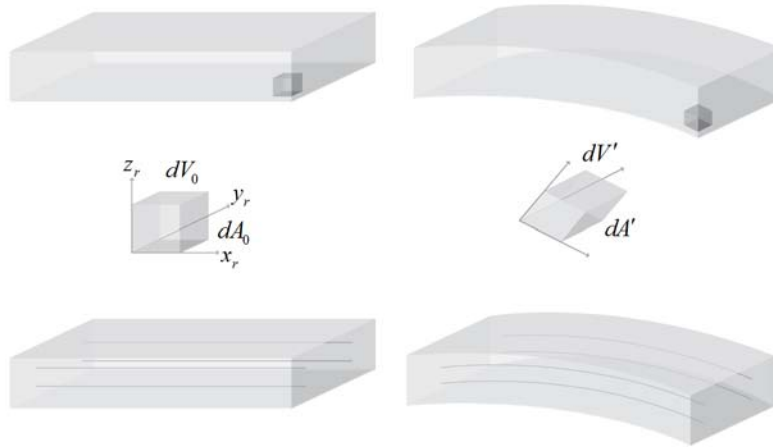
$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \mathbf{u} + I$$

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 2-14](#) 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 2-14: 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 2-14](#) to see the deformation of the  $X$ -axis). Particular care is therefore required when defining physics in this coordinate system.

#### FRAME CONTROL CONFLICTS

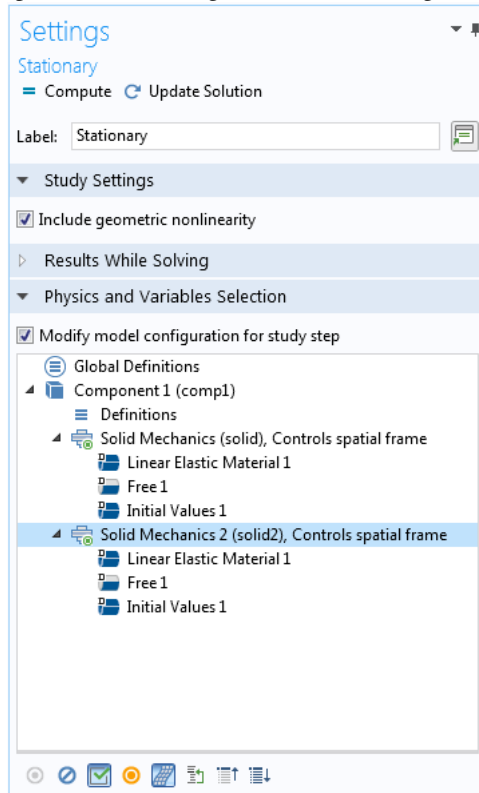
Under geometric nonlinearity, a Solid Mechanics interface (or a similar structural mechanics interface such as Multibody Dynamics) will assume control over the spatial frame in the domains where it is active. However, the definition of the spatial frame must be unique, and there may be conflicts. Some 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.



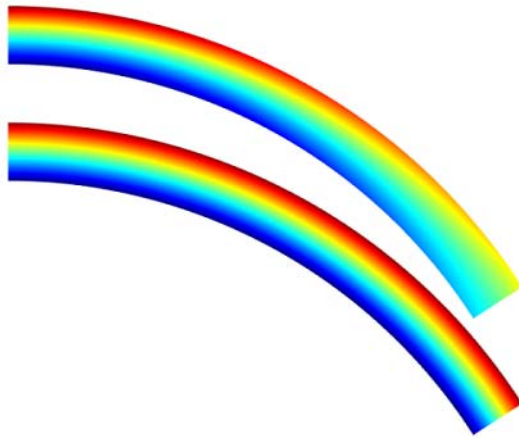
When you select a physics interface in the tree view, you can click the **Control Frame Deformation** button (  ) to toggle whether that interface should control the spatial frame or not.

## STRESS MEASURES

For example, in the spatial frame it is easy to define forces per unit 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 2-15:  $xx$ -components of the Cauchy stress tensor (top) and Second-Piola-Kirchhoff stress tensor (below) 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.

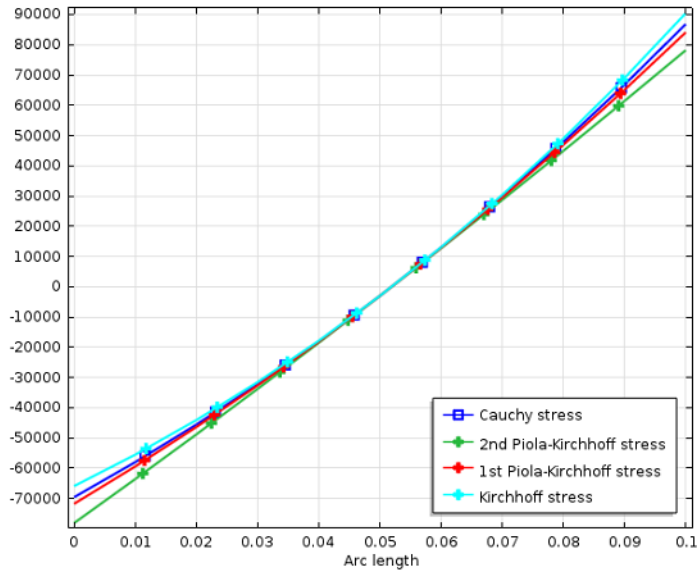


Figure 2-16: Stress distribution across the beam at the constrained end.

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.



### *Geometric Nonlinearity for the Solid Mechanics Interface*

---

For the Solid Mechanics interface, you enable a geometrically nonlinear analysis for a certain study step by selecting the **Include geometric nonlinearity** check box in the **Study Settings** section of a study step.

If any active feature in the model requires the analysis to be geometrically nonlinear, the **Include geometric nonlinearity** check box is selected automatically, and it cannot be cleared. The following physics features force this behavior:

- Contact, because the deformation between the contacting boundaries must be part of the solution.
- Moving mesh (when at least one deforming domain is active).
- Large strain plasticity.
- Hyperelastic materials, which are always formulated for large strains.

Usually you would also want to use geometric nonlinearity when a Moving Mesh interface is present, but this setting is not forced by the program.

When you select a geometrically nonlinear study step, the behavior of several features differs from that in a 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 either in 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 is changed from using engineering strains to Green-Lagrange strains, unless the **Force linear strains** check box is selected in the **Geometric Nonlinearity** section 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 actually are large.
- Pressure loads are interpreted as follower loads, so that the direction of the load as well as the loaded area are deformation dependent.
- 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, Beam, and Truss interfaces, a geometrically nonlinear analysis is enabled in the same way described above. For the Membrane interface, geometric nonlinearity must almost always be used, since it is nonlinear effects which supply the stiffness in the transverse direction.

The geometric nonlinearity in the Beam interface is limited to large rotations and displacements, but small strains.

The effect of using geometric nonlinearity in these interfaces is limited to the stress and strain representation as the other effects described in [Geometric Nonlinearity for the Solid Mechanics Interface](#) are not relevant.

### *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 too conservative solution strategy. 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 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 four predefined study sequences for prestressed dynamic analysis:

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

The prestressed study types assume that the loading causes small perturbations around the prestressed state.

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 which 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 however 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**.

### 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,  $\varepsilon_{ij}$ :

$$\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \sum_k \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) \quad (2-25)$$

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 electrical 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:

$$\begin{aligned} \mathbf{S} &= c_E \varepsilon - e^T \mathbf{E}_m \\ \mathbf{P}_m &= e \varepsilon + \varepsilon_0 (\varepsilon_{rS} - \mathbf{I}) \mathbf{E}_m \end{aligned}$$

where  $\mathbf{S}$  is the second Piola-Kirchhoff stress;  $\varepsilon$  is the Green-Lagrange strain,  $\mathbf{E}_m$  and  $\mathbf{P}_m$  are the electric field and electric polarization in the material orientation;  $\mathbf{I}$  is the identity matrix; and  $c_E$ ,  $e$ , and  $\varepsilon_{rS}$  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

$$\mathbf{D}_m = \mathbf{P}_m + \varepsilon_0 J C^{-1} \mathbf{E}_m$$

where  $C$  is the right Cauchy-Green tensor

$$C = F^T F$$

Fields in the global orientation result from the following transformation rules:

$$\begin{aligned} \mathbf{E} &= F^{-T} \mathbf{E}_m \\ \mathbf{P} &= J^{-1} F \mathbf{P}_m \\ \mathbf{D} &= J^{-1} F \mathbf{D}_m \\ \rho_v &= \rho_v J^{-1} \end{aligned} \tag{2-26}$$

where  $F$  is the deformation gradient;  $J$  is the determinant of  $F$ ; and  $\rho_v$  and  $\rho_v$  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

$$\begin{aligned} S &= c_E \varepsilon - e^T \mathbf{E}_m \\ \mathbf{D}_m &= e \varepsilon + \varepsilon_0 \varepsilon_{rS} \mathbf{E}_m + \varepsilon_0 (J C^{-1} - \mathbf{I}) \mathbf{E}_m \end{aligned}$$

### 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:

$$\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}$$



[The Electrostatics Interface](#) in the *COMSOL Multiphysics Reference Manual*

---

#### **LARGE DEFORMATION AND DEFORMED MESH**

The [Piezoelectricity 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 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 (<http://www.solidmechanics.org>), 2010.

# 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 boundaries which may come into contact, the *gap*. If the gap is positive then there is no interaction. The task of the contact algorithm 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](#)
- [Meshing for Contact Analysis](#)
- [Settings for Contact Nodes](#)
- [Quasistatic Contact Analysis](#)
- [Dynamic Contact Analysis](#)
- [Multiphysics Contact Analysis](#)
- [Special Types of Contact Problems](#)
- [Fallback Nodes for Contact](#)
- [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, 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.

To model a mechanical contact problem, 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 **Contact Pair** nodes under **Definitions**. A contact pair consists of two sets of boundaries, which are called the *source* and *destination* boundaries. Contact pairs can also be added automatically, based on boundary adjacency when the **Form an assembly** action is used. The geometric gap distance is a variable set up by the contact pair, which also define operators for mapping variables or expressions between the selected boundaries.
- Add **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
  - 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.



- Documentation of the [Contact](#), [Friction](#), [Slip Velocity](#), [Adhesion](#), [Decohesion](#), and [Wear](#) nodes.
- [Contact Analysis Theory](#).
- [Identity and Contact Pairs](#) in the *COMSOL Multiphysics Reference Manual*.

In a multiphysics analysis, a contact problem 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.



Because of the multiphysics capabilities, 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 of the contact properties are then made in the respective physics interfaces.

The fact that you add a **Contact** node to your model will automatically make all study steps geometrically nonlinear.

#### 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.



The **Friction** node is not available with the Layered Shell interface, and the **Slip Velocity** node is only available with the Solid Mechanics and Multibody Dynamics interfaces.

---

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.



The **Adhesion** and **Decohesion** nodes are not available with the Layered Shell interface.

---

### 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 done using two, fundamentally different, approaches.

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 the wear does not change the orientation of the normal to the boundary significantly.

Computing the amount of wear involves solving a rate equation, hence, it is only possible to compute wear in time-dependent study. The wear rate is typically a function of the contact pressure and the relative slip velocity between the contacting boundaries.



The **Wear** node is only available with the Solid Mechanics and Multibody Dynamics interfaces.

---

## SELECTING THE CONTACT ALGORITHM

In COMSOL Multiphysics, there are two classes of algorithms available for solving contact problems: the *penalty* method and the *augmented Lagrangian* method. For both algorithms, the contact pair is asymmetric, that is, the *destination* contact boundary is constrained not to penetrate the *source* boundary, but not vice versa.

### *Penalty Method*

The default penalty method is rather 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 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 tradeoff between accuracy and stability. While a large penalty factor will reduce nonphysical 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.

### *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 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.

### *Dynamic Methods*

Both the penalty and the augmented Lagrangian methods are additionally available in a specialized formulation intended for dynamic contact problems. Both of these are 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 formulations are viscous, the duration of the contact should be small, otherwise the overclosure will eventually grow and violate the non-penetration condition. This is especially the case for the dynamic penalty method, where the gap rate is only approximately zero.

### **CONTACT DETECTION**

The contact search is made only towards one side of the source and destination boundaries, determined by the positive direction of the contact normal of the selected boundary. For contact to be detected, this means that the source and destination contact normals must point towards each other. 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** node 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 outwards 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 (Solid Mechanics or Multibody Dynamics) defines a normal vector which 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.
- 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. If a Shell, Layered Shell or Membrane interfaces is an external source to a **Contact** node, the geometry normal of the boundary should point towards the destination boundary



When using the Layered Shell interface, it is recommended to set up the contact problem so that the contact surface is the ‘top’ side of the shell. Furthermore, the preferred position of the layered material is ‘Upside on boundary’.

---




The actual normal vector used in the contact search algorithm 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 the source and destination normals point towards each other.

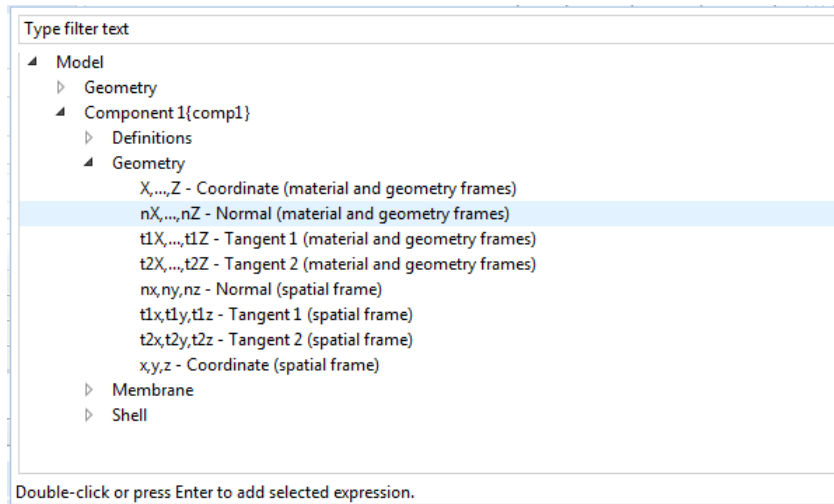
---

#### *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** check box 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.

### SOURCE SELECTION OUTSIDE PHYSICS INTERFACE

In most cases, the source and destination boundaries belong to the same physics interface. The only strict rule, however, is that the destination side belongs to the physics interface in which the **Contact** node resides. 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 the current physics interface, the gap is computed using only the current location of its mesh, ignoring any physical properties that may exist there. In this case, the **Contact** node 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 three main scenarios where you may want to use a source selection 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**.

### *Different Physics Interfaces*

In cases where you want to model contact between, for example, a 3D solid modeled in the Solid Mechanics interface, and a boundary modeled in the Shell interface, you can use this technique. In principle, you can put the **Contact** node in either of the physics interfaces, but it should not be used in both.

In practice, however, you will typically want to place the **Contact** node in the Shell interface, if that is one of the participating interfaces. The reason is that this is the only way to account for the actual shell thickness and possible offset in a contact analysis. When the source selection does not belong to the physics interface, only the location of the mesh is known when setting up the contact equations.

### *Contact Pairs*

---

#### **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 a **Rigid Domain** 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 the physics interface to which the **Contact** node is added, that must be the destination side.
- If you are modeling contact between a Shell interface and something else, the Shell interface should be on the destination side. Another way to state this is that the **Contact** node should be added to the Shell interface. The same principle applies to the Layered Shell and Membrane interfaces.

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.

When making the source selection, you can improve efficiency by selecting a region such that every point on the destination ‘sees’ a corresponding source point. The search follows the normal from the destination boundary. The search will be faster if a point on the source is found than if it is not, even if this point will never come into contact.

#### **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. When you add **Contact** nodes in the physics

interface, you select which of these suggested 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 material point on the destination boundary will see the same material 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 convergence to 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.

### *Meshing for Contact Analysis*

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 nonphysical 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 may 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 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 for both the penalty method and the augmented Lagrangian method, 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 higher value will decrease the unphysical penetration, but can be detrimental to the numerical conditioning of the stiffness matrix. A too small value can, however, 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 augmented Lagrangian 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 ( $E_{\text{equ}}$ ) of the material on the destination side. For linear elastic isotropic materials,  $E_{\text{equ}}$  is the actual Young’s modulus. For other types of materials,  $E_{\text{equ}}$  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),  $E_{\text{equ}}$  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.

#### *Trigger Cutback*

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 makes an attempt with a smaller time or parameter step. To speed up this process, you can select the **Trigger cutback** check box 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*

It is possible to assign an offset to both the source and destination boundaries. 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  $\Delta_{\text{gap}}$  are adjusted to be zero. By default,  $\Delta_{\text{gap}}$  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.

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 over-constraints 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 non-conforming. 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 a too high integration order can significantly increase the cost of assembling the matrices.



## 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.

## 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 to **Contact**. To model decohesion, it is required that an **Adhesion** node is present and active in the same parent **Contact** 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 post-buckling 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**  $\tau$ . 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  $\tau$  must be chosen with care. As a rule of thumb,  $\tau$  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 **Quasi static** 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, 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 Domain](#) 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.
- 

### *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 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 too weak spring will give a too large initial overclosure of the contact boundaries. A too stiff spring might influence the solution too much.

### *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 main interest is often the kinematics between the contacting bodies. If you rely on the (default) adaptive time-stepping algorithm, the solver typically also tries to resolve the wave propagation in the domains adjacent to the contact pairs. This can cause unnecessarily small time steps, and increase the computation time. To avoid this, you can modify the solver to use a manual time-stepping algorithm in the settings for the **Time-Dependent Solver** in the solver sequence. Make sure to use time steps that are small enough to capture the contact event, otherwise spurious energy contributions can result, and cause the problem to ‘blow up’.

The time-dependent solvers in COMSOL Multiphysics introduce numerical damping to stabilize the time stepping. This kind of stabilization is often necessary. However, excessive numerical damping runs the risk of removing vital information from the simulation. For this reason, the BDF solvers should be avoided for dynamic contact analyses. The default solver suggests the generalized- $\alpha$  solver when inertial terms are included in the structural mechanics problem.

Regardless of the method that is used, and how the solver is set up, it is good practice to do an *a posteriori* check of conservation of momentum and energy, to ensure that the solution is acceptable.



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.

### *Multiphysics Contact Analysis*

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 of 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.

### *Special Types of Contact Problems*

---

#### **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 in the in the settings for the **Contact** node. The offset should be defined by a parameter, so that the boundaries of the two domains are barely in contact in the initial state. Now, the offset can be reduced to zero step-by-step, using an auxiliary sweep in the solver.

#### *Nominal Geometry*

When working with a nominal geometry, you always need to add an offset in the **Contact** node. 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, 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 symmetric contact formulation, as the contact conditions are formulated on both sides of the contact pair.

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](#).

### *Fallback Nodes for Contact*

---

When a **Contact** node is added to a Solid Mechanics interface, it always contains one subnode, **Free**. This is the default fallback condition on the part of the boundary where contact is not established. For the Shell, Layered Shell and Membrane interfaces, no default fallback condition is added. By right-clicking the **Contact** node, you can add a new fallback condition. The most common case 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** fallback feature. The **Contact** node in the Layered Shell interface has no fallback features.

---

For a Solid Mechanics interface:

- If you add an ordinary **Boundary Load** node after a **Contact** node and use the same boundary selection, the load will be applied also to the part of the boundary that is in contact. In most cases, this is not a physically correct behavior.
- If a **Boundary Load** node appears before a **Contact** node in the model tree, the load on the common selection will be ignored, irrespective of whether the boundary actually is in contact or not.



For a Shell or Membrane interface:

- A **Face Load** node that have the same boundary selection as the **Contact** node is always applied to the entire boundary.
- 

### *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 use do so by setting the predictor to **Constant** when contact is present. However, it can sometimes be beneficial 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 too high penalty factor may cause the 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 too small penalty factor 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, 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 which is relevant for typical 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. Set the scale for each variable to a value that is representative for the expected result. Too large values may give a too early convergence, while too small values 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 an 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 non-converged 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 non-smooth, 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.

## DYNAMIC CONTACT

The dynamic contact methods in general inherit the properties and suggested solver settings from the corresponding standard method. With regards to solver settings, the main difference for the dynamic methods is how to set up the time-dependent solver, where it is recommended to use manual time stepping, see [Dynamic Contact Analysis](#).

### *Monitoring the Solution*

---

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** check box 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 if 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 2-11](#) 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 2-11: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
Tn	Contact pressure	The contact pressure in the normal direction	Contact: <b>Augmented Lagrangian</b> used	Lumped Step
Tt	Friction force	The friction force vector	Friction: <b>Augmented Lagrangian</b> used	Lumped Step



TABLE 2-11: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
gap_old	Previous gap distance	Physical gap variable from the last converged solution,	Contact: <b>Penalty, dynamic</b> or <b>Augmented Lagrangian, dynamic</b> 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: <b>Store accumulated slip</b> 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: <b>Add contact status to solver log</b> selected	Internal state variable
isSliding_old	Sliding friction status variable previous step	Nonzero if the point was in a sliding state at last converged solution	Contact: <b>Add contact status to solver log</b> 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: <b>Add contact status to solver log</b> selected. Friction node present.	Internal state variable

TABLE 2-11: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
Wfric	Frictional dissipation density	Energy dissipated by friction	Friction present and <b>Compute frictional dissipation</b> selected.	Segregated step
Wfric_old	Previous frictional dissipation density	Energy dissipated by friction at the last converged solution	Slip velocity present and <b>Compute frictional dissipation</b> 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 <b>Displacement-based damage</b> selected	Internal state variable
Ydm_max_old	Maximum damage energy at previous step	Maximum value of the stored undamaged elastic energy at previous step	Decohesion node present and <b>Energy-based damage</b> selected	Internal state variable

TABLE 2-11: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
vdmg_old	Viscous damage at previous step	Value of the viscous damage variable at the previous step	Decohesion node present and <b>Delayed damage</b> 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 <b>Compute damage dissipation energy</b> 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 2-12](#) 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 2-13](#) presents global variables related to contact that can be useful for postprocessing.

TABLE 2-12: 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
<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	

TABLE 2-12: IMPORTANT CONTACT VARIABLES

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
<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 2-13: 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
<phys>.Wfric_tot	Total frictional dissipation	Friction or Slip Velocity	Integration over all contact boundaries

TABLE 2-13: GLOBAL CONTACT VARIABLES

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
<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*

I. 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

When you want to be able to activate or deactivate one or several domains selected in a **Linear Elastic Material** node, add an **Activation** subnode. You then select the domains that you want to subject to **Activation**. 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 2-17](#) 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.

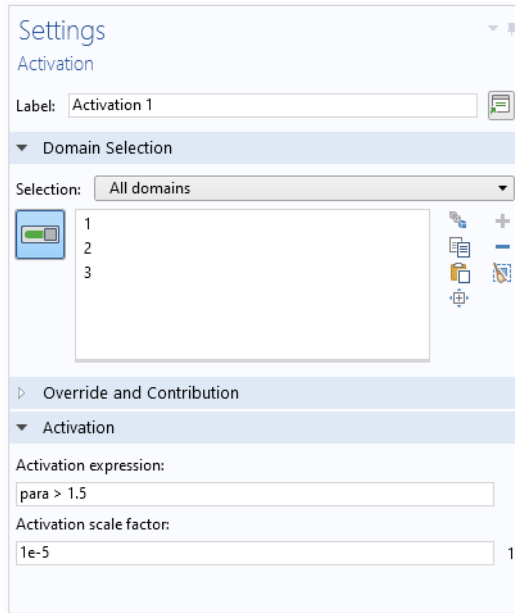


Figure 2-17: Settings for the Activation subnode.

## 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.

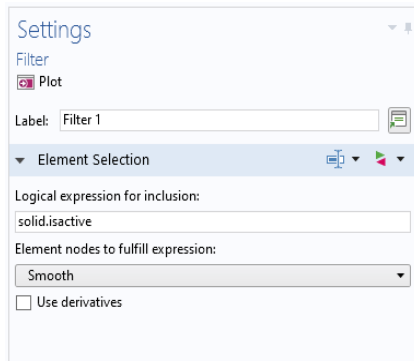
Domains that are not selected in any **Activation** subnode always have the `isactive` variable set to '1'.

TABLE 2-14: 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 2-18](#). 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 2-18: Filter settings to only display active material.*



# Springs and Dampers

The [Spring Foundation](#) and [Thin Elastic Layer](#) physics 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. This is a common problem in contact modeling before an assembly has actually settled. 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

$$\mathbf{f}_{s1} = (1 + i\eta)\mathbf{f}_s$$

where  $\mathbf{f}_s$  is the elastic spring force, and  $\eta$  is the loss factor.

Loss factor damping is only applicable for eigenfrequency and frequency-domain analysis. In 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 two-way 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  $T_{\text{ref}}$  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*  $\epsilon_{\text{th}}$ , which produces stress-free deformations. For a linear elastic material, the constitutive law is

$$\sigma = \mathbf{C} : (\epsilon - \epsilon_{\text{th}})$$



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

$$\frac{dL}{L} = \alpha_t(T)dT \quad (2-27)$$

where  $\alpha_t$  is the tangential thermal expansion coefficient. This form, which is the thermodynamic definition, is conceptually simple, because  $\alpha_t$  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  $\alpha$  will however depend on the choice of reference temperature,  $T_{\text{ref}}$ , 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 2-27 can be integrated, giving

$$\ln\left(\frac{L}{L_0}\right) = \int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau \quad (2-28)$$

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

$$\alpha(T, T_{\text{ref}}) = \frac{I(T, T_{\text{ref}})}{(T - T_{\text{ref}})} \quad (2-29)$$

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 2-28 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 2-27 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:

$$L(T) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \quad (2-30)$$

When using the measured data, it is possible that the strain-free state occurs at a temperature  $T_{\text{ref}}$  which differs from  $T_m$ . The dilation at any temperature  $T$  would then be defined as  $L(T) - L(T_{\text{ref}})$ , where  $L(T_{\text{ref}})$  can be written as.

$$L(T_{\text{ref}}) = [1 + \alpha_m(T_{\text{ref}})(T_{\text{ref}} - T_m)]L(T_m) \quad (2-31)$$

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  $L(T_{\text{ref}})$  can be related using [Equation 2-30](#) but with  $T_m$  replaced by  $T_{\text{ref}}$ .

$$L(T) = [1 + \alpha_r(T)(T - (T_{\text{ref}}))]L((T_{\text{ref}})) \quad (2-32)$$

Here  $\alpha_r(T)$  is the redefined thermal expansion coefficient, based on  $T_{\text{ref}}$ . It can be derived from the relations above. Using [Equation 2-30](#) and [Equation 2-32](#) there are two ways of writing the current length  $L(T)$ , so that

$$[1 + \alpha_r(T)(T - (T_{\text{ref}}))]L((T_{\text{ref}})) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \quad (2-33)$$

Equation 2-31 makes it possible to eliminate  $L(T_{\text{ref}})$  and  $L(T_m)$  from Equation 2-33, and after some algebra  $\alpha_r(T)$  can then be written as

$$\alpha_r(T) = \frac{\alpha_m(T) + (T_{\text{ref}} - T_m) \frac{\alpha_m(T) - \alpha_m(T_{\text{ref}})}{T - T_{\text{ref}}}}{1 + \alpha_m(T_{\text{ref}})(T_{\text{ref}} - T_m)} \quad (2-34)$$

#### 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 2-19. 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 2-19 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 2-34, but with a small modification to avoid problems if  $T = T_{\text{ref}}$ .

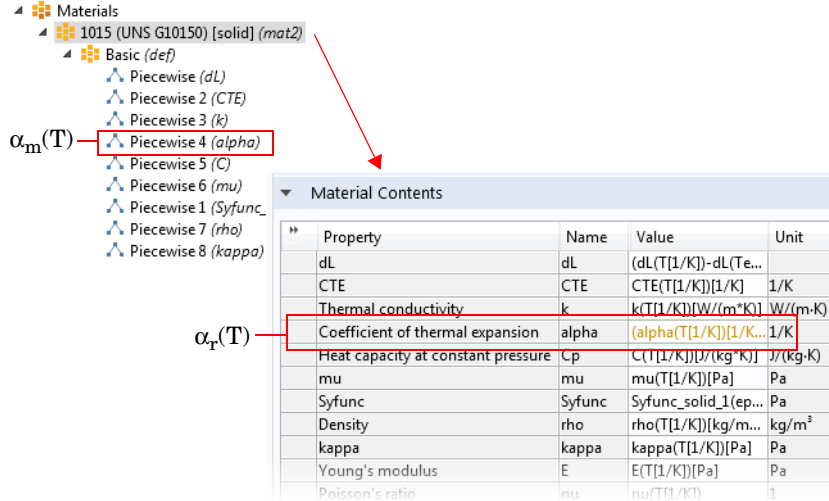


Figure 2-19: An example in COMSOL Multiphysics showing the Materials branch and where to find the temperature-dependent thermal expansion coefficient.

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 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 work.

- 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  $\alpha_m(T)$  in the function that you created.



### *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 Domains.

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 relieve 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 taken into account 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 four different structural mechanics interfaces: Solid Mechanics, Multibody Dynamics, Shell, and Membrane. The fluid-flow can be modeled with any physics interface from the **Single-Phase Flow** and **Two-Phase Flow, Phase Field** groups.

## *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 external 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 this 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 build-up 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

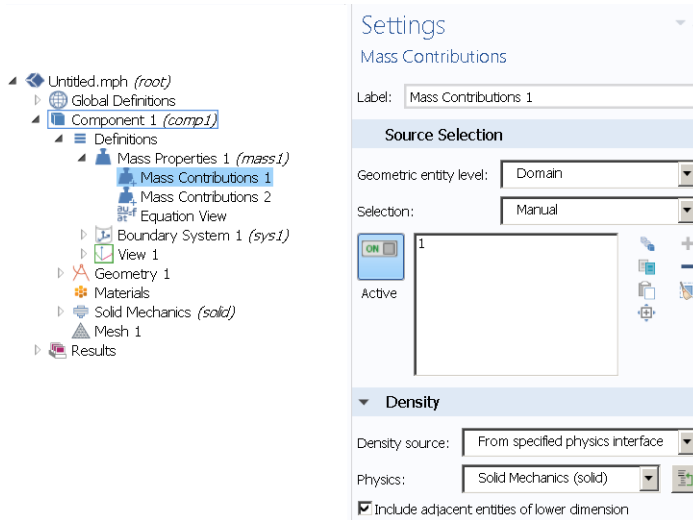
In the case of a fluid loading to structure coupling type, the structural mechanics problem can be treated as quasistatic. This can be handled by running the structural analysis as a parametric sweep over a number of static load cases, where time is used as the parameter.



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**.

# 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**. 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.

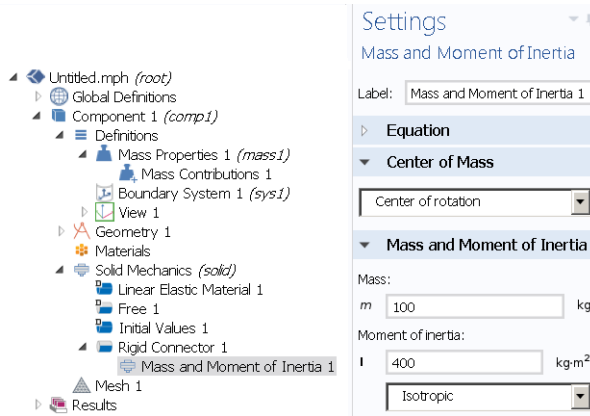
Variables			
Name	Expression	Unit	Description
mass1.volume	mass1.intDim3(root.round(mass1.con...)	m <sup>3</sup>	Volume
mass1.mass	mass1.intDim0(mass1.rho_pnt)+mass...	kg	Mass
mass1.CMX	(mass1.intDim0(mass1.rho_pnt*X)+...	m	Center of mass, X component
mass1.CMY	(mass1.intDim0(mass1.rho_pnt*Y)+...	m	Center of mass, Y component
mass1.CMZ	(mass1.intDim0(mass1.rho_pnt*Z)+...	m	Center of mass, Z component
mass1.IXX	mass1.intDim0(mass1.rho_pnt*(Y^2...)	kg·m <sup>2</sup>	Moment of inertia XX component
mass1.IXY	mass1.intDim0(-mass1.rho_pnt*X*Y)...	kg·m <sup>2</sup>	Moment of inertia XY component
mass1.IYX	mass1.IXY	kg·m <sup>2</sup>	Moment of inertia YX component
mass1.IXZ	mass1.intDim0(-mass1.rho_pnt*X*Z)...	kg·m <sup>2</sup>	Moment of inertia XZ component
mass1.IZX	mass1.IXZ	kg·m <sup>2</sup>	Moment of inertia ZX component
mass1.IYY	mass1.intDim0(mass1.rho_pnt*(X^2...)	kg·m <sup>2</sup>	Moment of inertia YY component
mass1.IYZ	mass1.intDim0(-mass1.rho_pnt*Y*Z)...	kg·m <sup>2</sup>	Moment of inertia YZ component
mass1.IZY	mass1.IYZ	kg·m <sup>2</sup>	Moment of inertia ZY component
mass1.IZZ	mass1.intDim0(mass1.rho_pnt*(X^2...)	kg·m <sup>2</sup>	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 Domain, 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** check box 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** check box 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 Domain** 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.

## *RVE Modeling Using the Cell Periodicity Node*

---

To model an RVE, you add the domain feature **Cell Periodicity**, and select the domains representing the RVE. For each pair of matching boundaries, add a **Boundary Pair** subnode, and select the boundaries.

In principle, there is no limitation on the physics features you can use for modeling the RVE, as long as the basic assumptions about periodicity are not violated. You should, however, not add any displacement constraints because the possible rigid-body motions are automatically constrained by the **Cell Periodicity** node.

### **MESHING**

The accuracy of the analysis is significantly better if each pair of matching boundaries has the same mesh. Mesh the boundaries before the domains, and use **Copy Face** to ensure that the boundary meshes match.

If you decide to use a nonmatching mesh, then the stress disturbances can be reduced by using a weak form of the periodicity constraints.

## 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 unit rise in temperature. Similarly, for hygroscopic properties, add **Hygroscopic Swelling** subnodes and apply a unit rise 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. 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** and **Average stress** periodic conditions, you can calculate the homogenized elastic properties of the RVE. Use the average strain condition to compute the elasticity matrix, or the 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 perturbation 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 created load group. If 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.

For the **Average strain** periodic condition, you can also create a global material using the **Create Material** option under the **Study and Material Generation** button. This option creates a material that contains the homogenized elasticity 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 to a user-defined material library, you should use the numerical values presented under **Results**. The automatically generated global material is linked to the **Cell Periodicity** node in the current model.



*Micromechanical Model of a Composite:* Application Library path

**Structural\_Mechanics\_Module/Material\_Models/**

**micromechanical\_model\_of\_a\_composite**

---

# 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 designed to simplify such analyses.

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.

## MODELING THE BOLTS

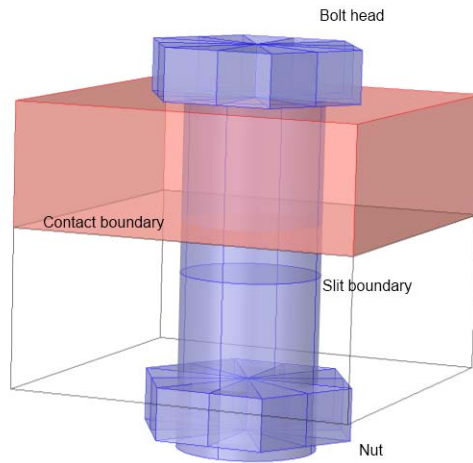
You must use a specific modeling technique in order to use a bolt in a prestress analysis.

- 1 Model each bolt 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 parallel to the  $Z$ -axis.
- 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 2-20). The slit boundary can be composed of several 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** check box in the **Boundary Selections** section of the settings for the part instance (Figure 2-21).
- 4 If needed, add contact conditions between the bolt head and the component and between different components clamped by the bolt.
- 5 Add a **Bolt Pretension** node, in which the pretension force or stress is prescribed for a set of bolts with the same data.

- 6 For each bolt having an identical pretension stress, add a **Bolt Selection** subnode where the slit boundary is selected.



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, not as the force in the modeled half. 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 2-20: Example of a bolted joint.*

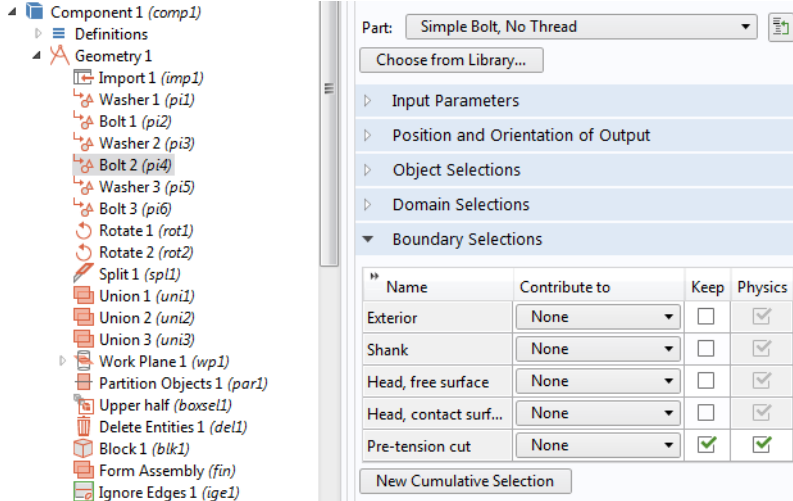


Figure 2-21: Getting access to the slit boundary selection for a bolt from the Part Libraries.

Each bolt defined in the **Bolt Selection** node has a single 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:

$$\mathbf{u}_u = \mathbf{u}_d - (d - r)\mathbf{n}$$

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.
- 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, this action is not 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.

---

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. Also, in versions prior to 5.3, this automatic mechanism was not available, so in older models the studies were always set up manually.



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** check box.

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.pb1t1.sb1t1.F_bolt`. The bolt results are summarized in the table below.

TABLE 2-15: 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.



[Studies and Solvers](#) in the *COMSOL Multiphysics Reference Manual*



*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.

In the **Bolt Thread Contact** node, you enter general data about the bolt geometry. For each individual bolt, you need to add one **Thread Boundary Selection** subnode.

## **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.

# 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 post-failure 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 nodes of the beam elements. This means that higher-order variations of the displacement field between mesh nodes 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 high to accurately enforce the regularized constraint, but if they are too high the overall stiffness matrix will be ill-conditioned. This means that in most cases, the spring stiffness is 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 2-16](#). All spring stiffness components have the same default expressions.

TABLE 2-16: DEFAULT EXPRESSION FOR SPRING STIFFNESS.

CONNECTION TYPE	SPRING CONSTANT PER UNIT LENGTH	SPRING CONSTANT PER UNIT SURFACE AREA
Truss	$1e3 * \text{truss.Eequ} * \text{truss.area} / h^2$	$1e3 * \text{truss.Eequ} * \text{truss.perimeter} / h^2$
Beam	$1e3 * \text{beam.Eequ} * \text{beam.area} / h^2$	N/A
Membrane	N/A	$1e5 * \text{mbrn.Eequ} * \text{mbrn.d} / h^2$

### 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 interface only the direction of the local edge tangent  $\mathbf{t}_{1e}$  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  $\mathbf{t}_{1e}$  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 global  $z$ -direction. Hence, the first transverse direction is

$$\mathbf{t}_1 = \frac{\mathbf{e}_z - (\mathbf{t}_{1e} \cdot \mathbf{e}_z)\mathbf{t}_{1e}}{\|\mathbf{e}_z - (\mathbf{t}_{1e} \cdot \mathbf{e}_z)\mathbf{t}_{1e}\|}$$

and the second transverse direction is

$$\mathbf{t}_2 = \mathbf{t}_{1e} \times \mathbf{t}_1 = \frac{\mathbf{t}_{1e} \times \mathbf{e}_z}{\|\mathbf{e}_z - (\mathbf{t}_{1e} \cdot \mathbf{e}_z)\mathbf{t}_{1e}\|}$$

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  $\mathbf{u}_{1,p}$ , the accumulated slip  $u_{pe}$ , and the friction dissipation density  $W_p$ . The last variable is only added if the **Calculate dissipated energy** check box 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  $c = c_0 + c_h$ , 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 nevertheless 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  $0.1 * \langle \text{tag} \rangle . \text{ExtCp10p}(\text{solid.p})$  to define  $c_0$  with a friction coefficient equal to 0.1. Here  $\langle \text{tag} \rangle$  is the tag of the multiphysics coupling and  $\langle \text{tag} \rangle . \text{ExtCp10p}(\text{epxr})$  is the operator that maps expression form 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.



# 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 nomenclature, this is called a *slit condition*.
- By its true 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 internal 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.

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 which 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.

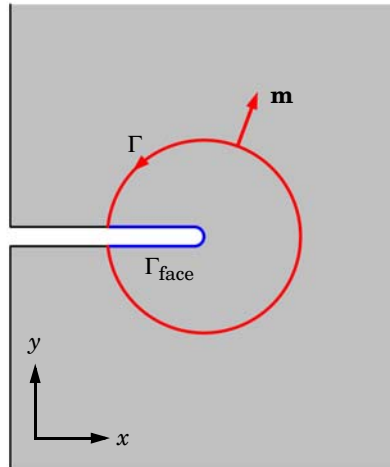


Figure 2-22: Example of a J-integral path.

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.
- 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**.

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 internal or external 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.

#### **DEFAULT PLOTS**

When J-integral nodes are present, they will generate additional default plots. 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 2-23](#).

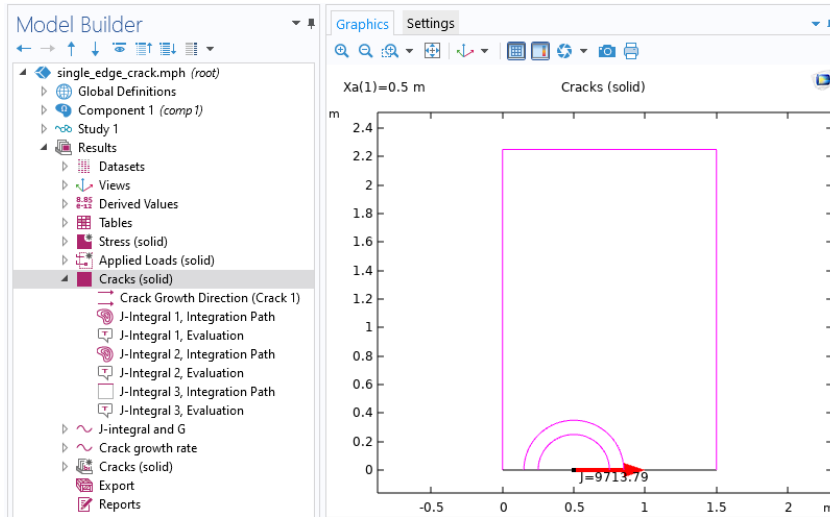


Figure 2-23: Default plot for a 2D case with a crack in a symmetry plane. Three integration paths are shown; 2 circular and one along boundaries.

### 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 2-24](#).

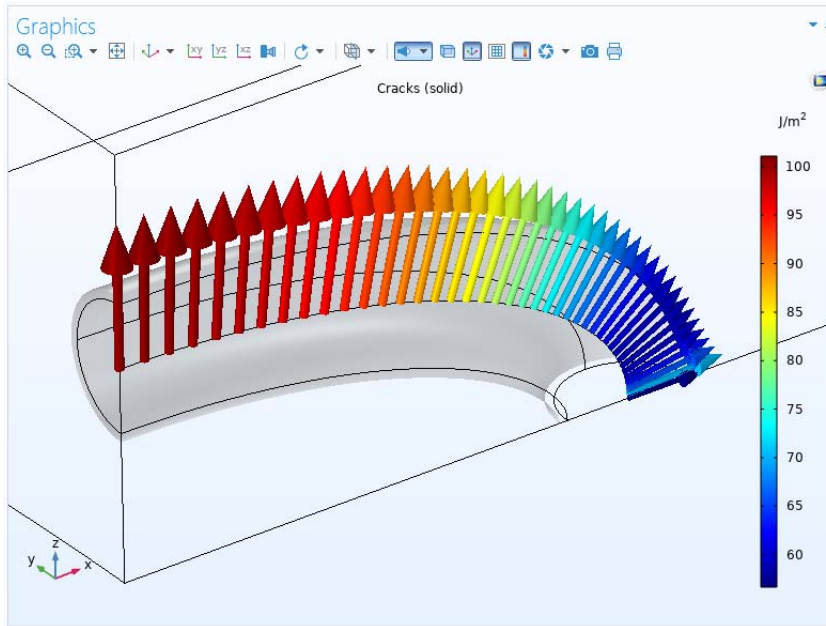


Figure 2-24: Default plot for a case with a 3D crack in a symmetry plane. One integration path is used.

#### DEFAULT EVALUATION GROUPS

In both 3D and 2D, a default evaluation group, named **Cracks**, 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.

#### *Stress Intensity Factors*

Stress intensity factors are computed from the J-integral. 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:

$$J = \frac{1}{E_{\text{eff}}} \left( K_{\text{I}}^2 + K_{\text{II}}^2 + \frac{1}{1-\nu} K_{\text{III}}^2 \right) \quad (2-35)$$

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 2-35,  $E_{\text{eff}}$  is an effective Young's modulus, defined as

$$E_{\text{eff}} = \begin{cases} \frac{E}{1-\nu^2} & \text{if plane strain} \\ E & \text{if plane stress} \end{cases}$$

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 however be noted that for most metals, the variation in  $E_{\text{eff}}$  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  $\text{N}/\text{m}^{3/2}$ . 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  $\text{N}/\text{m}^{3/2}$ .

# 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 the 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 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 write expressions that are functions of the spatial coordinates. The expression for the load inside the physics interface can however 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 a general such situation, then 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, and is thus a type of modal superposition. It 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.



Three studies are added:

- One study with an **Eigenfrequency** study step for computing the eigenfrequencies and corresponding eigenmodes.
- One study with a **Frequency Domain** study step. This study can mainly be considered as a placeholder which is a mandatory input to the model reduction.
- One study with a **Model Reduction** study step, in which the ROM is created. It references the two previous studies. In the common case that you have already computed the eigenmodes, you can change the setting of the **Training study** in this node to point to the old eigenfrequency study, and then delete the newly generated one.

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 gravity load. This is how vibration testing is analyzed.

- 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  $\log(\text{PSD})$  values. You then reference this function through an expression like `exp(int1(log(freq)))`. 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.

The evaluation of random vibration results is done through special operators listed in [Table 2-17](#) which are defined by the **Random Vibration** node (with a tag `rvib1` in this example).

TABLE 2-17: RANDOM VIBRATION OPERATORS

OPERATOR	SYNTAX	EXPLANATION
<code>psd</code>	<code>rvib1.psd(exp)</code>	Evaluates the PSD of an expression for the given frequency
<code>cross</code>	<code>rvib1.cross(exp1, exp2)</code>	Evaluates the cross-correlation between two expressions for a given frequency,
<code>rms</code>	<code>rvib1.rms(expr, f1, f2, N)</code>	Evaluates the RMS value of an expression within the frequency interval [f1, f2]. N points are used for integrating over the interval.

TABLE 2-17: RANDOM VIBRATION OPERATORS

OPERATOR	SYNTAX	EXPLANATION
m	<code>rvibl.m(K,expr, f1, f2, N)</code>	Evaluates the K:th order moment of the PSD. Thus, <code>rvibl.rms(expr, f1, f2, N) = sqrt(rvibl.m(0,expr, f1, f2, N))</code>
q2	<code>rvibl.q2(expr, f1, f2, N)</code>	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 <code>rvibl.q2(utt^2,...)</code> , you can write <code>rvibl.q2(u^2*(2*pi*freq)^4,...)</code> .  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.

It is convenient to plot the PSD of a result quantity as a function of frequency 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.



- *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 which 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:

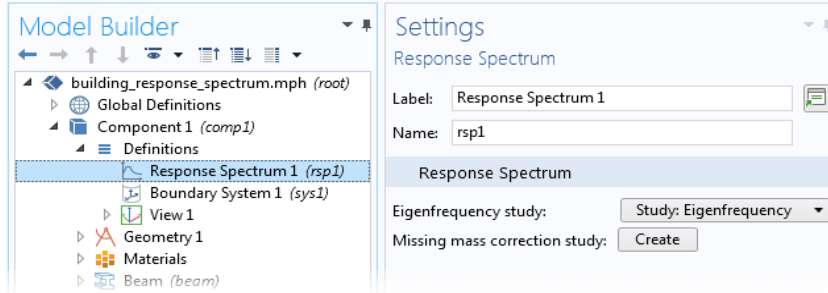
- An eigenfrequency solution, where the eigenmodes to be included in the response spectrum analysis have been computed. All computed eigenmodes are used. If you want to filter out a particular set of modes, you can add a **Combine Solutions** study step after the **Eigenfrequency** study step. Such a 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 **Missing mass correction study: Create** button in the **Response Spectrum** node.

#### *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

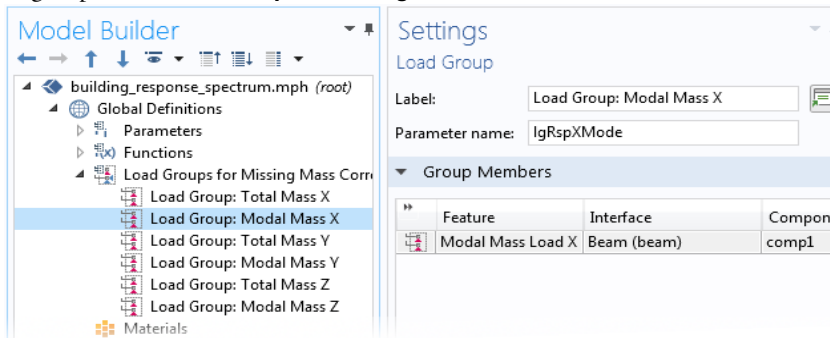
You create a missing mass correction study from the **Response Spectrum** node by clicking the **Missing mass correction study: Create** button.



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**.

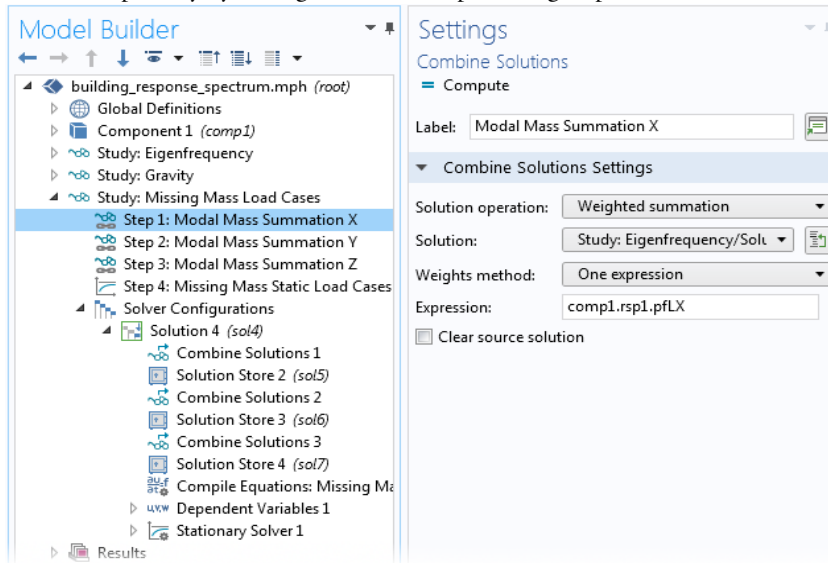


The **Parameter name** of the load group is reserved, and you should not modify it.

- A new study named **Study: Missing Mass Load Cases** is created. This study contains three or four study steps, depending on the spatial 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.



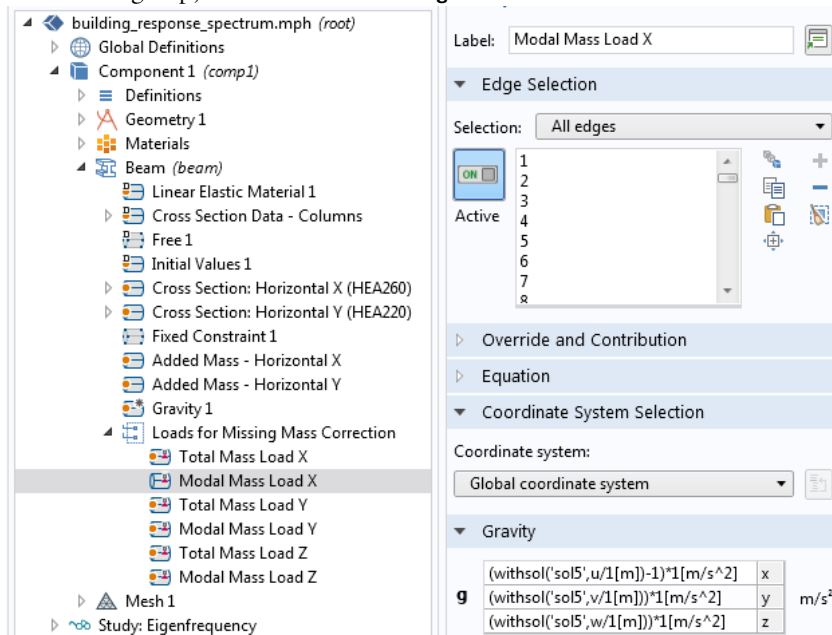
If the physics interface has loads other than those automatically generated for missing mass correction, you need to make sure that those loads are not solved for in the **Missing Mass Static Load Cases** study step. There are several ways of doing that. You can, for example:



- 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 they 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** check box. 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, since it is only compatible with the assumptions about how the rigid modes are scaled.

### **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 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**

# 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.

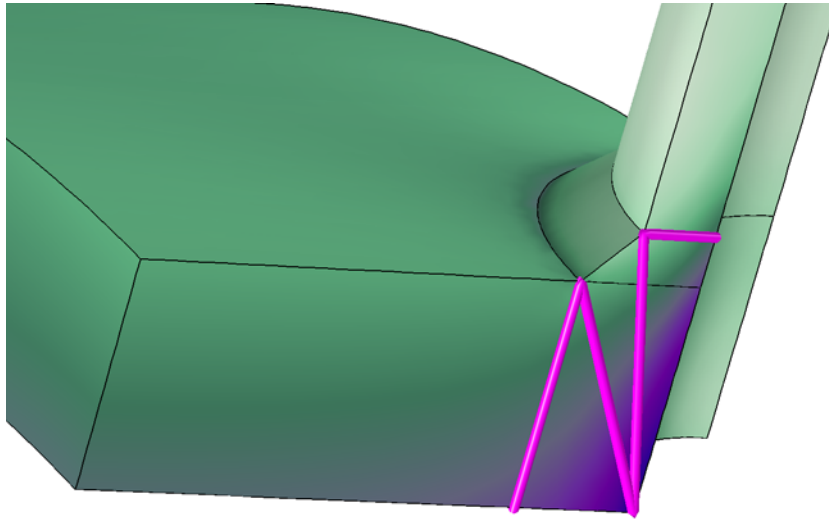
## ORIENTATION OF STRESS COMPONENTS

The stresses along the SCL are represented in a local orthonormal coordinate system,  $x_1$ - $x_2$ - $x_3$ . The  $x_1$  direction is oriented along the SCL, so it is mainly the stresses in the second and third directions which are of interest.

- In 3D, you must specify the  $x_2$  direction, and thus implicitly the  $x_3$  direction. You specify the orientation either by selecting a point in the  $x_1$ - $x_2$  plane, or by defining an orientation vector in an approximate  $x_2$  direction. In either case, the actual  $x_2$  direction is chosen so that it is perpendicular to the SCL, and lies in the plane you have specified. The  $x_3$  orientation is then taken as perpendicular to  $x_1$  and  $x_2$ . As long as you are only interested in a stress intensity, the choice of orientation is arbitrary.
- In 2D, the  $x_3$  direction is the out-of-plane direction, and the  $x_2$  direction is perpendicular to the SCL in the  $XY$ -plane.
- In 2D axial symmetry, the  $x_3$  direction is the azimuthal direction, and the  $x_2$  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.



*Figure 2-25: Four stress classification lines in a transition region at a pressure vessel nozzle.*

There are, however, some situations where the 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 any longer be possible to use swept meshes, or the mesh quality is reduced in critical regions.

In this case, a possible solution is to operate on a copy of the geometry in another component. It is then possible to use submodeling — that is, to study only a local region with its boundaries having displacements controlled by the solution from the larger model.

Below is an outline of the steps you need to take for this approach.

- I In the original component (assume that its tag is `comp1`), add a **General Extrusion** operator. Set **Source frame** to **Material**. You can name the operator, but in the

following description, the default name `genext1` is assumed. This operator will be used for mapping results to the second component.

- 2 Add a new component of the same dimension as the one in which you performed the original analysis.
- 3 Add the geometry — using, for example, **Insert Sequence** or **Import** — to the new component (`comp2`).
- 4 Add the lines you are going to use for stress linearization to the imported geometry.
- 5 Add a Solid Mechanics interface, and select only the domains which contain the new lines. You can even change the geometry so that you cut out parts of the domains.
- 6 Add a **Prescribed Displacement** node having domain selection. Select **All domains**. Prescribe the displacement in all directions to be the same as in the original model with expressions like `comp1.genext1(u)` for **Prescribed in x direction**.
- 7 If the original analysis contains inelastic strains, such as thermal expansions, these must also be included. You can do this either by adding a **Thermal Expansion** node with appropriate settings, or by explicitly importing the inelastic strains using an **External Strain** node. In the latter case, you would use expressions like `comp1.genext1(solid.eth11)` for the strain components.
- 8 Add **Stress Linearization** nodes for the new linearization lines.
- 9 Add the materials that were used in `comp1`. The most efficient approach is to add them under **Global Definitions** and link to the same material definitions from both components.
- 10 Create a mesh for the domains in `comp2` that you are solving for. It is only the mesh close to the new edges to which you need to pay any attention.
- 11 Add a new stationary study, in which you solve only for the new Solid Mechanics interface in `comp2`.
- 12 In the settings for the new study set **Values of variables not solved for** to point to the solution from which you want to pick the results in `comp1`. You can also add an **Auxiliary sweep**, if the original analysis contains results for several parameters or time steps.
- 13 Run the new study.

## 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 default plot are generated.

One edge dataset is created for each SCL. These datasets are named **Linearization Line <n>**, where n is an integer number.

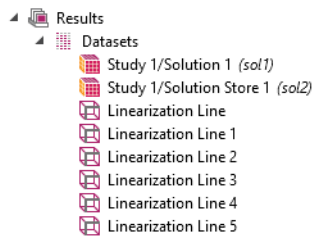


Figure 2-26: Generated datasets in a model with five SCL.

The default 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.

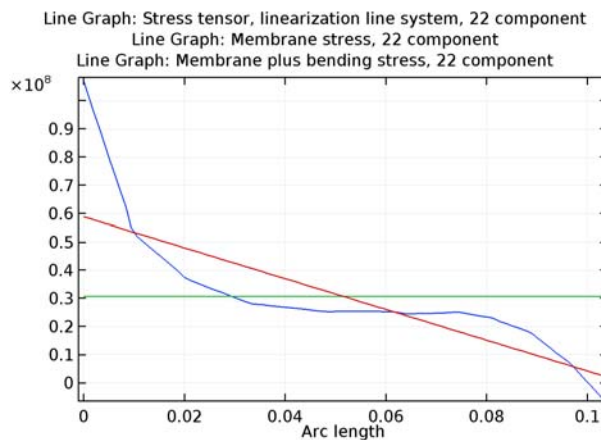


Figure 2-27: Default plot along a stress classification line.

## 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, 's11'.

As an example, the variable `solid.Sm22` and the variable `solid.s11.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 2-18: VARIABLES FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Sllj	Stress tensor in local coordinate system	$\sigma_{ij}$	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	$\sigma_{p(end), ij}$	ij = 11, 12, 13, 22, 23, 33	Feature
Slm	Stress intensity, membrane	$\sigma_{int}$		Physics, Feature
Slmbs	Stress intensity, membrane + bending, starting point	$\sigma_{int}$		Feature
Slbme	Stress intensity, membrane + bending, endpoint	$\sigma_{int}$		Feature

TABLE 2-18: VARIABLES FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Slmb	Stress intensity, membrane + bending,		max(Slmb <sub>s</sub> , Slmb <sub>e</sub> )	Physics, Feature
N <sub>ij</sub>	Local in-plane force	$N_{ij}$	$ij = 22, 23, 33$	Feature
M <sub>ij</sub>	Local bending moment	$M_{ij}$	$ij = 22, 23, 33$	Feature
Q <sub>i</sub>	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

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 Solution Components](#)



[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 on 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 a MUMPS direct solver in 2D and PARDISO direct solver in 3D. For large 3D problems (several hundred thousands 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, 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 presmoothing and postsmoothing can give better results compared to SOR that is the default for 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 is however 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 Solution Components*

---

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, then 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.
- If there are other types of physics interfaces being solved in the same study, then the segregated solver should usually be kept. Make sure that all structural mechanics degrees of freedom except 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 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.s112` — 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.S112` — 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.e112` — 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. Under **Definitions>Variable Utilities**, add a **Vector Transform** or **Matrix Transform** 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**.

- 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.

- In the **Vector Transform** or **Matrix Transform** node, select the local coordinate system in the **Output** section. In the latter node, select it twice.
- Select the domain where you want this transformation to be valid.
- Often, you want to enter your own name for the transformed object in the **Name** text field.

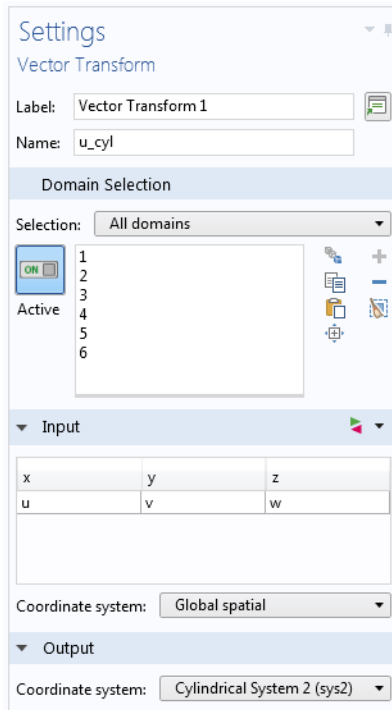


Figure 2-28: Example of settings for transforming the displacement vector to a cylindrical coordinate system.

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.

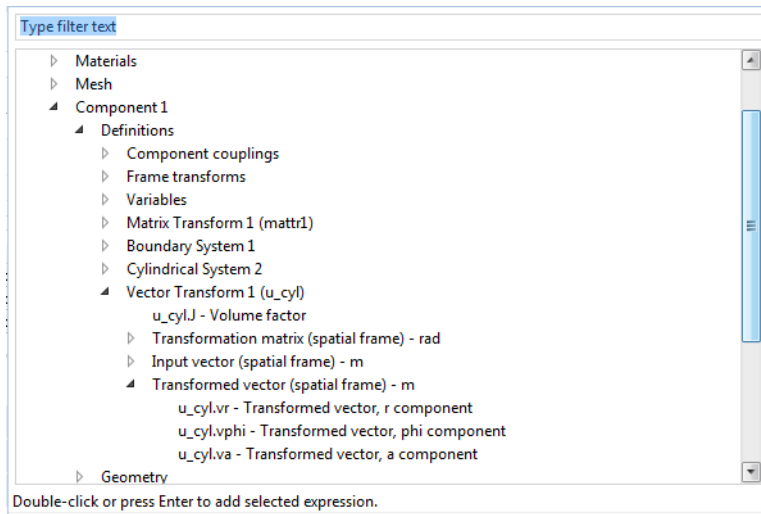


Figure 2-29: Picking a transformed result in the Replace Expression dialog.



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  $\sigma$  is then computed as

$$\sigma = \sigma_{ex} + C : \epsilon_{el} = \sigma_{ex} + C : (\epsilon - \epsilon_{inel})$$

where  $C$  is the 4th order elasticity tensor, “:” stands for the double-dot tensor product (or double contraction). The elastic strain  $\epsilon_{el}$  is the difference between the total strain  $\epsilon$  and all inelastic strains  $\epsilon_{inel}$ . In this example, the inelastic strain tensor has two contributions, creep strain  $\epsilon_{cr}$  and thermal strain  $\epsilon_{th}$ . There may also be an extra stress term  $\sigma_{ex}$  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  $\epsilon$  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  $\epsilon_{th}$  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  $\epsilon_{cr}$  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  $\epsilon_{inel}$  (in this example, the creep strains is now subtracted from the total strains, to form the elastic strain  $\epsilon_{el}$ ).
- The elastic strain  $\epsilon_{el}$  is multiplied by the elasticity tensor  $C$ , evaluated at  $\mathbf{x}$ , to give the stress  $\sigma$ . The material properties may depend on the location, either explicitly, or for example by a temperature dependency.
- Any extra stress  $\sigma_{ex}$ , 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 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 a too low resolution in the mesh. As soon as you are investigating the quality of the solution, it is recommended to switch off any smoothing.

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 with an artificial variation.

The default resolution depends on several factors, but will always imply that a result quantity is evaluated in a rather large number of points in the element.

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 is to create a low order field that approximates the ‘wavy’ expression over the element, as described in the next section.



Entering Quality Settings for Plot Settings Windows 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. This operator has, in its most basic version, the 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 a least squares fit. If there are state variables stored at Gauss points, as is the case with many nonlinear material models, it is important that you use the correct integration order in the operator in order to evaluate those variables accurately. In most cases, when quadratic shape functions are used, this means that the first argument should be 4.

For some common cases, physics interfaces define built-in variables with Gauss point evaluation. As an example, the built-in variable `solid.sGpx` is equivalent to using the expression `gpeval(4, solid.sx)`.

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 thus exists in the domain and can be evaluated from it 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 the line, using Gauss points on the element edges. This is probably not what you want, in particular if the state variables are stored at Gauss points in the domain. In order to avoid this, create domain variables under **Definitions>Variables** using the operator syntax.

### **EIGENMODE SCALING**

The eigenmodes that form the solution to eigenvalue problems, like eigenfrequency and linear buckling, have some special properties which 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** check box 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 of 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 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 sets 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 2-30: 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, default plots containing the applied loads are generated. Figure 2-31 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.

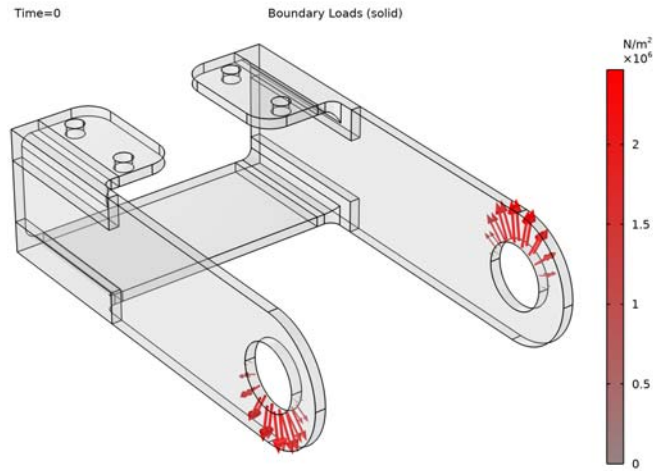


Figure 2-31: Example of an automatically generated plot to visualize the boundary loads applied to a bracket.

### 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 2-19.

TABLE 2-19: PLOT GROUP NAMES AND PLOT TYPES

LOAD GROUP NAME	PLOT TYPE	DESCRIPTION
Volume Loads	Arrow Volume, Arrow Surface, Arrow Line	Generated from <b>Body Load</b> , <b>Gravity</b> , and <b>Rotating Frame</b> . The plot type depends on the geometrical dimension of the physics interface.
Boundary Loads	Arrow Surface	Generated from <b>Boundary Load</b> . Also, from <b>Added Mass</b> on a boundary combined with <b>Gravity</b> or <b>Rotating Frame</b> .
Face Loads	Arrow Surface	Generated from <b>Face Load</b> in Shell, Plate, and Membrane interfaces.

TABLE 2-19: PLOT GROUP NAMES AND PLOT TYPES

LOAD GROUP NAME	PLOT TYPE	DESCRIPTION
Face Moments	Arrow Surface	Generated from <b>Face Load</b> in Shell, Plate, and Membrane interfaces if nonzero moments are present.
Edge Loads	Arrow Line	Generated from <b>Edge Load</b> . Also, from <b>Added Mass</b> on an edge combined with <b>Gravity</b> or <b>Rotating Frame</b> .
Edge Moments	Arrow Line	Generated from <b>Edge Load</b> in Shell, Plate, and Beam interfaces if nonzero moments are present.
Point Loads	Arrow Point	Generated from <b>Point Load</b> .
Point Moments	Arrow Point	Generated from <b>Point Load</b> in Shell, Plate, and Beam interfaces if nonzero moments are present.
Global Loads, Global Moments	Point Trajectories	Generated from global features like <b>Rigid Connector</b> and <b>Rigid Domain</b> , which have subnodes as <b>Applied Load</b> and <b>Applied Moment</b> . 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 <b>Fluid Load</b> in Pipe Mechanics interface.
Fluid Loads (Drag Force)	Arrow Line	Generated from <b>Fluid Load</b> 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** check box 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 2-31](#). 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 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 non-constant.
- 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 requested by the currently active features in all physics interfaces active in the study.

Note that the default 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, either delete the current dataset and its associated plot groups, or create a new solution sequence. Then compute the solution or do the **Get Initial Value** to generate the load plots.

# 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 [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.

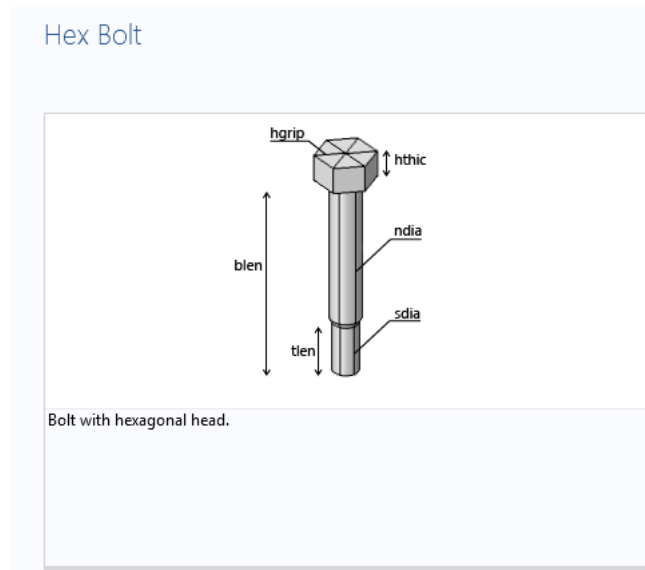
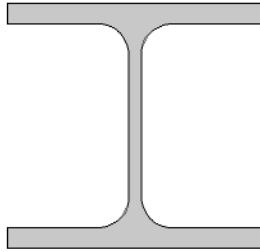


Figure 2-32: Example of a bolt part.

- 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.

### European HEA-beam

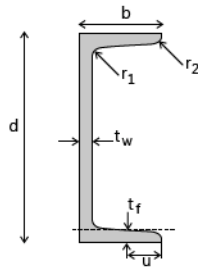


European wide flange beams.

A single size parameter controls all section dimensions. For example, entering 120 as parameter gives a standard HEA 120 beam cross section.

Figure 2-33: Example of a standard beam cross section.

### Generic C-beam



Generic channel cross-section shape with either flat or tapered flange.

Figure 2-34: 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*

---



# Structural Mechanics Theory

This chapter contains the theory behind the implementation of the structural mechanics functionality in COMSOL Multiphysics. It is assumed that the reader has a basic knowledge of solid mechanics.

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 Boundary Conditions](#)
- [Stress Linearization](#)
- [J-integral Theory](#)

- Embedded Elements
- Average Rotation
- Energy Quantities

# Solid Mechanics Theory

## *Introduction*

---

In the following, the theory for the Solid Mechanics interface is described. To a large extent, this theory covers other structural mechanics physics interfaces, such as Shell and Beam, which are included with the Structural Mechanics Module. For these other interfaces, only the details which are specific to a certain interface are described its documentation.



- [Theory for the Shell and Plate Interfaces](#)
  - [Theory for the Membrane Interface](#)
  - [Theory for the Beam Interface](#)
  - [Theory for the Beam Cross Section Interface](#)
  - [Theory for the Truss Interface](#)
- 

### **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

$$\sigma = C:\varepsilon$$

Here, the stress tensor  $\sigma$  and the strain tensor  $\varepsilon$  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

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$$

where the Einstein summation convention has been used as a shorthand for

$$\sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 C_{ijkl}\varepsilon_{kl}$$

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

$$\sigma^{ij} = C^{ijkl} \varepsilon_{kl}$$

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 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  $r$ ,  $\varphi$ , 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  $R$ ,  $\Phi$ , 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

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \quad (3-1)$$

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  $\mathbf{X}$ . 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.

# 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](#)
- [Axial Symmetry and Deformation](#)
- [Generalized Plane Strain](#)
- [Out-of-Plane and Circumferential Modes](#)

## *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

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$$

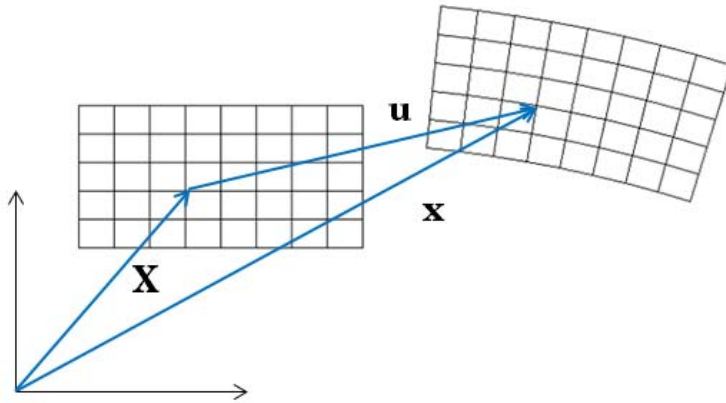
Here,  $\mathbf{x}$  is the *spatial coordinate* and  $\mathbf{X}$  is the *material coordinate*.



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.

$$\nabla = \nabla_X = \begin{bmatrix} \frac{\partial}{\partial X} & \frac{\partial}{\partial Y} & \frac{\partial}{\partial Z} \end{bmatrix}$$

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,  $\mathbf{F}$ , shows how an infinitesimal line element,  $d\mathbf{X}$ , is mapped to the corresponding deformed line element  $d\mathbf{x}$  by

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} d\mathbf{X} = \mathbf{F} d\mathbf{X}$$

The deformation gradient  $\mathbf{F}$  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,  $\mathbf{F}$  can be written as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \mathbf{u} + \mathbf{I}$$

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

$$\frac{dV}{dV_0} = \frac{\rho_0}{\rho} = \det(\mathbf{F}) = \mathcal{J}$$

Here,  $\rho_0$  is the initial density and  $\rho$  is the current density after deformation. The determinant of the deformation gradient tensor  $\mathbf{F}$  is related to volumetric changes with respect to the initial state. A pure rigid body displacement implies  $\mathcal{J} = 1$ . Also, an incompressible material is represented by  $\mathcal{J} = 1$ . These are called *isochoric* processes.

The determinant of the deformation gradient tensor is always positive (since a negative mass density is unphysical). The relation  $\rho = \rho_0/\mathcal{J}$  implies that for  $\mathcal{J} < 1$  there is compression, and for  $\mathcal{J} > 1$  there is expansion. Since  $\mathcal{J} > 0$ , the deformation gradient  $\mathbf{F}$  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. You should thus 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 you add [Thermal Expansion \(for Materials\)](#) to the material model.



The variable `solid.rho` represents a “reference” or “initial” density  $\rho_0$ , and not the “current” density  $\rho$ . 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  $\mathbf{F}$  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

$$\mathbf{F} = \mathbf{R}\mathbf{U}$$

where  $\mathbf{R}$  is a proper orthogonal tensor ( $\det(\mathbf{R}) = 1$ , and  $\mathbf{R}^{-1} = \mathbf{R}^T$ ) and  $\mathbf{U}$  is the *right stretch tensor* given in the material frame. The rotation tensor  $\mathbf{R}$  describes the rigid rotation, and all information about the deformation of the material is contained in the symmetric tensor  $\mathbf{U}$ .



- 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  $U$  tensor are the principal stretches,  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ . 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  $\varepsilon_{\text{eng}}$  is the engineering strain. The three principal stretches act along three orthogonal directions. In the coordinate system defined by these principal directions, the  $U$  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

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \cdot \frac{\partial u_k}{\partial X_j} \right). \quad (3-2)$$



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 the 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.e111`, `solid.e112`, and so on.
- In a geometrically linear analysis, the strain variables `solid.eX`, `solid.e111`, and so on, will instead represent engineering strain.
- The right Cauchy–Green deformation tensor in local coordinate system are named `solid.C111`, `solid.C112`, 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 ( $\varepsilon_{xy}$ ,  $\varepsilon_{yz}$ ,  $\varepsilon_{xz}$ ) is used.



In the documentation, the symbol  $\varepsilon$  is used to denote strain in general. In a geometrically nonlinear analysis, the strain should be interpreted as a Green–Lagrange strain. In a geometrically linear analysis, the engineering strain is used.

The symmetric strain tensor  $\varepsilon$  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

$$\varepsilon_r = \frac{\partial u}{\partial r}, \quad \varepsilon_\varphi = \frac{u}{r}, \quad \varepsilon_z = \frac{\partial w}{\partial z}, \quad \text{and} \quad \gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}$$

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 and Deformation](#).

## 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  $\lambda$  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

$$\varepsilon_{1p} = \begin{bmatrix} \log\lambda_1 & 0 & 0 \\ 0 & \log\lambda_2 & 0 \\ 0 & 0 & \log\lambda_3 \end{bmatrix}$$

This tensor is then transformed to the global coordinate system in order to give the logarithmic strain tensor.

## *Invariants of Strain*

---

### PRINCIPAL STRAINS

The principal strains are the eigenvalues of the strain tensor ( $\varepsilon$ ), 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 cases. The corresponding vectors in the principal directions,  $\mathbf{v}_{p_i}$ , 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.

In a geometrically nonlinear analysis, the Green–Lagrange strain tensor is used. 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

$$I_1(\boldsymbol{\varepsilon}) = \text{trace}(\boldsymbol{\varepsilon})$$

$$I_2(\boldsymbol{\varepsilon}) = \frac{1}{2}(\text{trace}(\boldsymbol{\varepsilon})^2 - \text{trace}(\boldsymbol{\varepsilon}^2))$$

$$I_3(\boldsymbol{\varepsilon}) = \det(\boldsymbol{\varepsilon})$$

The invariants of the strain deviator tensor is also useful.



$$\begin{aligned}\operatorname{dev}(\boldsymbol{\varepsilon}) &= \boldsymbol{\varepsilon} - \frac{1}{3}\operatorname{trace}(\boldsymbol{\varepsilon})\mathbf{I} \\ \mathcal{J}_1(\boldsymbol{\varepsilon}) &= \operatorname{trace}(\operatorname{dev}(\boldsymbol{\varepsilon})) = 0 \\ \mathcal{J}_2(\boldsymbol{\varepsilon}) &= \frac{1}{2}\operatorname{dev}(\boldsymbol{\varepsilon}) : \operatorname{dev}(\boldsymbol{\varepsilon}) = \frac{1}{3}I_1(\boldsymbol{\varepsilon})^2 - I_2(\boldsymbol{\varepsilon}) \\ \mathcal{J}_3(\boldsymbol{\varepsilon}) &= \det(\operatorname{dev}(\boldsymbol{\varepsilon})) = \frac{2}{27}I_1(\boldsymbol{\varepsilon})^3 - \frac{1}{3}I_1(\boldsymbol{\varepsilon})I_2(\boldsymbol{\varepsilon}) + I_3(\boldsymbol{\varepsilon})\end{aligned}$$

As defined above  $\mathcal{J}_2 \geq 0$ .  $I_1$  represents the relative change in volume for infinitesimal strains and  $\mathcal{J}_2$  represents the magnitude of shear strain.

In tensor component notation, the invariants can be written as

$$\begin{aligned}\operatorname{trace}(\boldsymbol{\varepsilon}) &= \varepsilon_{kk} \\ \operatorname{dev}(\boldsymbol{\varepsilon})_{ij} &= \varepsilon_{ij} - \frac{\varepsilon_{kk}}{3}\delta_{ij} \\ I_2(\boldsymbol{\varepsilon}) &= \frac{1}{2}(\varepsilon_{ii}\varepsilon_{jj} - \varepsilon_{ij}\varepsilon_{ji}) \\ \mathcal{J}_2(\boldsymbol{\varepsilon}) &= \frac{1}{2}\operatorname{dev}(\boldsymbol{\varepsilon})_{ij}\operatorname{dev}(\boldsymbol{\varepsilon})_{ji}\end{aligned}$$

The volumetric strain is defined as

$$\varepsilon_{\text{vol}} = \operatorname{trace}(\boldsymbol{\varepsilon})$$

and the equivalent deviatoric strain as

$$\varepsilon_{\text{deve}} = \sqrt{\frac{2}{3}\operatorname{dev}(\boldsymbol{\varepsilon})_{ij}\operatorname{dev}(\boldsymbol{\varepsilon})_{ji}} = \sqrt{\frac{4}{3}\mathcal{J}_2(\boldsymbol{\varepsilon})}$$



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.e1dev11`, etc. The internal variable for the equivalent deviatoric strain is `solid.edeve`.

In terms of the principal strains, the strain invariants can be written as

$$\begin{aligned}
I_1(\boldsymbol{\varepsilon}) &= \varepsilon_{p1} + \varepsilon_{p2} + \varepsilon_{p3} \\
I_2(\boldsymbol{\varepsilon}) &= \varepsilon_{p1}\varepsilon_{p2} + \varepsilon_{p2}\varepsilon_{p3} + \varepsilon_{p1}\varepsilon_{p3} \\
I_3(\boldsymbol{\varepsilon}) &= \varepsilon_{p1}\varepsilon_{p2}\varepsilon_{p3}
\end{aligned}$$

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

$$L = \frac{d\mathbf{F}}{dt} \mathbf{F}^{-1}$$

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

$$\frac{d\boldsymbol{\varepsilon}}{dt} = \mathbf{F}^T \mathbf{L}_d \mathbf{F}$$

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,  $\boldsymbol{\varepsilon}_0$
- External strain,  $\boldsymbol{\varepsilon}_{ex}$
- Thermal strain,  $\boldsymbol{\varepsilon}_{th}$
- Hygroscopic strain,  $\boldsymbol{\varepsilon}_{hs}$
- Plastic strain,  $\boldsymbol{\varepsilon}_{pl}$
- Creep strain,  $\boldsymbol{\varepsilon}_{cr}$
- Viscoplastic strain,  $\boldsymbol{\varepsilon}_{vp}$

#### **ADDITIVE DECOMPOSITION**

In a geometrically linear analysis, the elastic strain is computed by a straightforward subtraction of the inelastic strain:

$$\boldsymbol{\varepsilon}_e = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{inel}$$

where

$$\boldsymbol{\varepsilon}_{inel} = \boldsymbol{\varepsilon}_0 + \boldsymbol{\varepsilon}_{ex} + \boldsymbol{\varepsilon}_{th} + \boldsymbol{\varepsilon}_{hs} + \boldsymbol{\varepsilon}_{pl} + \boldsymbol{\varepsilon}_{cr} + \boldsymbol{\varepsilon}_{vp}$$

Additive decomposition of strains can also be used in a geometrically nonlinear analysis. In this case, it can however only be justified as long as the strains are small. In the case of large deformations, the different strain contributions may not even be commutative.

You can choose to use additive decomposition also for geometric nonlinearity by selecting the **Additive strain decomposition** check box in the settings for [Linear Elastic Material](#) or [Nonlinear Elastic Material](#).

### MULTIPLICATIVE DECOMPOSITION

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.

The total deformation gradient tensor is defined as the result of two successive operations, an inelastic deformation followed by an elastic deformation:

$$\mathbf{F} = \mathbf{F}_{\text{el}}\mathbf{F}_{\text{inel}} \quad (3-4)$$



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  $\mathbf{F}$  tensor is defined by the displacements as usual and describes the mapping from the material frame to the spatial frame. The  $\mathbf{F}_{\text{inel}}$  tensor, however, describes a mapping from the material frame to an intermediate frame, and the  $\mathbf{F}_{\text{el}}$  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

$$\mathbf{F}_{\text{el}} = \mathbf{F}\mathbf{F}_{\text{inel}}^{-1} \quad (3-5)$$

so the inelastic deformations are removed from the total deformation gradient tensor. The *elastic right Cauchy–Green* deformation tensor is then computed from  $\mathbf{F}_{\text{el}}$ .

$$C_{\text{el}} = F_{\text{el}}^T F_{\text{el}}$$

and the *elastic Green–Lagrange* strain tensor is computed as:

$$\varepsilon_{\text{el}} = \frac{1}{2}(C_{\text{el}} - I)$$

The inelastic deformation tensor  $F_{\text{inel}}$  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  $F_{\text{inel}}$ .

$$F_{\text{inel}} = F_1 F_2 F_3 F_4 \dots$$

where  $F_i$  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 internal variables for the elastic right Cauchy–Green deformation tensor in the local coordinate system are named `solid.Ce111`, `solid.Ce112`, and so on; and for the elastic Green–Lagrange tensor in local coordinates `solid.ee111`, `solid.ee112`, and so on.

The elastic, inelastic, and total volume ratios are related as

$$\det(F) = \det(F_{\text{cl}})\det(F_{\text{inel}}) \text{ or } J = J_{\text{cl}}J_{\text{inel}}$$



The internal variables for the elastic, inelastic, and total volume ratio are named `solid.Je1`, `solid.Ji`, and `solid.J`.

### Large strain plasticity

In case of large strain plasticity, the plastic strains are primarily not represented as strains, but as the plastic deformation gradient tensor,  $F_{pl}$ .

The *plastic Green–Lagrange strain* tensor is computed from the *plastic deformation gradient* tensor as

$$\boldsymbol{\varepsilon}_{pl} = \frac{1}{2}(F_{pl}^T F_{pl} - I)$$

As opposed to the small strain formulation, the total, plastic, and elastic Green–Lagrange strain tensors are related as

$$\boldsymbol{\varepsilon}_{el} = F_{pl}^{-T}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{pl})F_{pl}^{-1}$$

### Axial Symmetry and Deformation

---

The axially symmetric geometry uses a cylindrical coordinate system. Such a coordinate system is orthogonal but curvilinear, and you can choose between a covariant basis  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$  and a contravariant basis  $\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3$  when formulating the theory.

The metric tensor is

$$[g_{ij}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in the coordinate system given by  $\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3$ , and

$$[g^{ij}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^{-2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ .

The metric tensor plays the role of a unit tensor for a curvilinear coordinate system.

For any vector or tensor  $A$ , the metric tensor can be used for conversion between covariant, contravariant, and mixed components:

$$A_i^j = \sum_m (A_{im} g^{mj})$$

$$A^{ij} = \sum_{m,n} (A_{nm} g^{ni} g^{mj})$$

In both covariant and contravariant basis, the base vector in the azimuthal direction has a non-unit length. To cope with this issue, the so-called physical basis vectors of unit length are introduced. These are

$$\mathbf{e}_r = \mathbf{e}_1 = \mathbf{e}^1, \mathbf{e}_\phi = \frac{1}{r} \mathbf{e}_2 = r \mathbf{e}^2, \mathbf{e}_z = \mathbf{e}_3 = \mathbf{e}^3$$

The corresponding components for any vector or tensor are called physical.

For any tensor, the physical components are defined as

$$A_{ij}^{\text{phys}} = \sqrt{g_{ii}} \sqrt{g_{jj}} A^{ij}$$

where no summation is done over repeated indices.

#### **MIXED COMPONENTS AND PRINCIPAL INVARIANTS**

The mixed tensor components are given by

$$A_j^i = \sum_m (g^{im} A_{mj})$$

The principal invariants are

$$I_1(A) = \text{trace}[A_i^i] = \sum_i A_i^i = A_{11} + A_{22} \frac{1}{r} + A_{33}$$

$$I_2(A) = \frac{1}{2} \left( (I_1(A))^2 - \sum_{i,j} A_j^i A_i^j \right)$$

$$I_3(A) = \det[A_i^i]$$

#### **DISPLACEMENTS AND AXIAL SYMMETRY ASSUMPTIONS**

The axial symmetry implementation in COMSOL Multiphysics assumes independence of the angle, and also that the azimuthal component of the displacement is identically zero. The physical components of the radial and axial displacement,  $u$  and  $w$ , are used as dependent variables for the axially symmetric geometry.

## STRAINS

The right Cauchy–Green deformation tensor is defined as

$$\mathbf{C} = ((\nabla \mathbf{u})^T + \nabla \mathbf{u} + (\nabla \mathbf{u})^T \nabla \mathbf{u}) + \mathbf{g}$$

and the Green–Lagrange strain tensor is

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\mathbf{C} - \mathbf{g})$$

Under the axial symmetry assumptions, the covariant components of  $\mathbf{C}$  are

$$C_{11} = 2\frac{\partial u}{\partial r} + \left[ \left( \frac{\partial u}{\partial r} \right)^2 + \left( \frac{\partial w}{\partial r} \right)^2 \right] + 1$$

$$C_{12} = C_{23} = 0$$

$$C_{13} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} + \left[ \frac{\partial u}{\partial r} \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \frac{\partial w}{\partial z} \right]$$

$$C_{22} = 2ru + [(u)^2] + r^2$$

$$C_{23} = r\frac{\partial v}{\partial z} + \left[ -v\frac{\partial u}{\partial z} + u\frac{\partial v}{\partial z} \right]$$

$$C_{33} = 2\frac{\partial w}{\partial z} + \left[ \left( \frac{\partial u}{\partial z} \right)^2 + \left( \frac{\partial w}{\partial z} \right)^2 \right] + 1$$

For geometrically linear analysis, drop the nonlinear terms inside square brackets in the above expressions.

The physical components of  $\boldsymbol{\varepsilon}$  are

$$\varepsilon_r = \varepsilon_{11} = \frac{1}{2}(C_{11} - 1)$$

$$\varepsilon_{r\phi} = \varepsilon_{\phi z} = 0$$

$$\varepsilon_{rz} = \varepsilon_{13} = \frac{1}{2}C_{13}$$

$$\varepsilon_\phi = \frac{\varepsilon_{22}}{r^2} = \frac{1}{2r^2}(C_{22} - r^2)$$



$$\varepsilon_z = \varepsilon_{33} = \frac{1}{2}(C_{33} - 1)$$

The volumetric strain is

$$\varepsilon_{\text{vol}} = I_1(\varepsilon) = \varepsilon_r + \varepsilon_\phi + \varepsilon_z$$

### *Generalized Plane Strain*

---

One possible extension of the plane strain formulation is to assume that the strains are independent of the out-of-plane coordinate  $z$ ; that is,

$$\varepsilon_{ij} = \varepsilon_{ij}(x, y)$$

Under the small strain assumption, the above equations have the following 3D solution:

$$\begin{aligned} u &= u_0(x, y) - \frac{a}{2}z^2 \\ v &= v_0(x, y) - \frac{b}{2}z^2 \\ w &= (ax + by + c)z \end{aligned}$$

where  $a$ ,  $b$ , and  $c$  are constants. Thus, at the cross section  $z = 0$ , one has

$$\begin{aligned} u &= u_0(x, y) \\ v &= v_0(x, y) \\ w &= 0 \end{aligned}$$

and

$$\begin{aligned} \varepsilon_z &= ax + by + c \\ \varepsilon_{xz} &= \varepsilon_{yz} = 0 \end{aligned}$$

The above conditions differ from the plane strain state only by the fact that the normal out-of-plane strain component can vary linearly throughout the cross section. This 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.

At the cross section  $z = 0$ , the deformation is in-plane and fully characterized by the in-plane displacement components  $u(x,y)$  and  $v(x,y)$ .

In COMSOL Multiphysics, the coefficients  $a$ ,  $b$ , and  $c$  in the expression for the  $\epsilon_z$  strain are modeled as extra degrees of freedom that are constant throughout the model (global variables).

### GEOMETRIC NONLINEARITY

In case of geometric nonlinearity, the strains are represented by the Green–Lagrange strain tensor:

$$\boldsymbol{\epsilon} = \frac{1}{2}[(\nabla \mathbf{u})^T + \nabla \mathbf{u} + (\nabla \mathbf{u})^T \nabla \mathbf{u}]$$

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 strain tensor expression and dropping quadratic and higher order terms in the coefficients, one obtains:

$$\begin{aligned} \epsilon_{XX} &= u_X + \frac{1}{2}(u_X^2 + v_X^2) \\ \epsilon_{YY} &= v_Y + \frac{1}{2}(u_Y^2 + v_Y^2) \\ \epsilon_{ZZ} &= a(X + u) + b(Y + v) + c \\ \epsilon_{XY} &= \frac{1}{2}(u_Y + v_X + u_X u_Y + v_X v_Y) \\ \epsilon_{XZ} &= \epsilon_{YZ} = 0 \end{aligned}$$

Thus, in the leading order approximation, the strains become independent of the out-of-plane coordinate  $Z$ .

**OUT-OF-PLANE WAVES**

If 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 represents a static in-plane prestress solution:

$$\mathbf{u}_0 = \begin{bmatrix} u_0(X, Y) \\ v_0(X, Y) \\ 0 \end{bmatrix}$$

This can be obtained by a standard static analysis using 2D geometry for the cross section with the corresponding boundary conditions.

The second part of the solution presents a time-harmonic linear perturbation with an amplitude that can be a function of the in-plane coordinates. Such a perturbation can be seen as an out-of-plane wave, with a small amplitude that propagates in the  $Z$  direction, and has a wavelength  $L$  and phase velocity  $c$ :

$$\begin{aligned} k_Z &= 2\pi/L \\ c &= \omega/k_Z \\ \mathbf{u}_1(X, Y) \exp\left[\frac{2\pi i}{L}(ct - Z)\right] \end{aligned}$$

Note that in contrast to the prestress, the perturbation amplitude can have nonzero values in all three 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, and thus the wave number  $k_Z$ , can be considered as a parameter. Then,  $\omega$  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

$$\begin{aligned}\omega &= 2\pi f \\ f &= f(L, \mathbf{u}_0) \\ c(\lambda, \mathbf{u}_0) &= Lf(L, \mathbf{u}_0)\end{aligned}$$

Thus, the wave speed for the out-of-plane wave is computed as a function of the wavelength and possible prestress in the material. The dependence of the wave speed on the wavelength is often called dispersion.

Alternatively, the frequency  $f$  (and thus  $\omega$ ) 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,

$$\begin{aligned}k_Z &= 2\pi/L \\ L &= L(f, \mathbf{u}_0) \\ c(f, \mathbf{u}_0) &= fL(f, \mathbf{u}_0)\end{aligned}$$

which determine the wavelength and speed for the wave that can propagate out-of-plane for a given frequency under given in-plane prestress condition. Such interpretation of the perturbation solution is sometimes called a signaling problem.

### CIRCUMFERENTIAL MODES

A standard 2D axially symmetric representation of the structure geometry assumes zero twist displacement component together with independence of the circumferential position  $\phi$ . The following 3D solution form represent an extension of these assumptions:

$$\mathbf{u}(R, \phi, Z, t) = \mathbf{u}_0(R, Z) + \mathbf{u}_1(R, Z) \exp(i\omega t - im\phi)$$

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. Thus,

$$\mathbf{u}(R, \phi + 2\pi, Z, t) = \mathbf{u}(R, \phi, Z, t)$$

The circumferential wave number can be introduced as  $k_m = m/R$ .

The static prestress solution  $\mathbf{u}_0$  has zero twist component

$$\mathbf{u}_0 = \begin{bmatrix} u_0(R, Z) \\ 0 \\ w_0(R, Z) \end{bmatrix}$$

and it can be obtained using a standard static analysis in 2D axially symmetric geometry.

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  $m$ , it can be found using an eigenfrequency analysis in a 2D axially symmetric geometry. Hence,

$$\begin{aligned} \omega &= 2\pi f \\ f &= f(m, \mathbf{u}_0) \end{aligned}$$

and the perturbation solution becomes

$$\mathbf{u}_1(R, Z) \exp[2\pi i f(m, \mathbf{u}_0)t - im\phi]$$

This represents eigenmodes in the corresponding 3D structure, which can be computed assuming certain constraints on the axis and possible static prestress without twist and independent off the position along the axis.

# Stresses

In this section:

- [Defining Stress](#)
- [Invariants of the Stress Tensor](#)
- [Plane Strain and Plane Stress Cases](#)
- [Initial Stresses and Strains](#)
- [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  $\sigma$  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  $\sigma$  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 be the one which you may consider as “largest”.

The convention used in [Ref. 1](#) 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

$$\begin{aligned} I_1(\sigma) &= \text{trace}(\sigma) \\ I_2(\sigma) &= \frac{1}{2}(\text{trace}(\sigma)^2 - \text{trace}(\sigma^2)) \\ I_3(\sigma) &= \det(\sigma) \end{aligned} \tag{3-6}$$

In many cases, the invariants of the deviatoric stress tensor are also useful.

$$\begin{aligned}
 \text{dev}(\boldsymbol{\sigma}) &= \boldsymbol{\sigma} - \frac{1}{3}\text{trace}(\boldsymbol{\sigma})\mathbf{I} \\
 J_1(\boldsymbol{\sigma}) &= \text{trace}(\text{dev}(\boldsymbol{\sigma})) = 0 \\
 J_2(\boldsymbol{\sigma}) &= \frac{1}{2}\text{dev}(\boldsymbol{\sigma}) : \text{dev}(\boldsymbol{\sigma}) = \frac{1}{3}I_1(\boldsymbol{\sigma})^2 - I_2(\boldsymbol{\sigma}) \\
 J_3(\boldsymbol{\sigma}) &= \det(\text{dev}(\boldsymbol{\sigma})) = \frac{2}{27}I_1(\boldsymbol{\sigma})^3 - \frac{1}{3}I_1(\boldsymbol{\sigma})I_2(\boldsymbol{\sigma}) + I_3(\boldsymbol{\sigma})
 \end{aligned} \tag{3-7}$$

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

$$\begin{aligned}
 \text{trace}(\boldsymbol{\sigma}) &= \sigma_{kk} \\
 \text{dev}(\boldsymbol{\sigma})_{ij} &= \sigma_{ij} - \frac{\sigma_{kk}}{3}\delta_{ij} \\
 I_2(\boldsymbol{\sigma}) &= \frac{1}{2}(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji}) \\
 J_2(\boldsymbol{\sigma}) &= \frac{1}{2}\text{dev}(\boldsymbol{\sigma})_{ij}\text{dev}(\boldsymbol{\sigma})_{ji}
 \end{aligned}$$

The pressure is defined as

$$p = \frac{-\text{trace}(\boldsymbol{\sigma})}{3} = \frac{-I_1(\boldsymbol{\sigma})}{3}$$

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.p**.

### 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)

$$\sigma_p^3 - I_1\sigma_p^2 + I_2\sigma_p - I_3 = 0$$

#### OTHER STRESS INVARIANTS

It is possible to define other invariants in terms of the primary invariants. One common auxiliary invariant is the *Lode angle*  $\theta$ .

$$\cos 3\theta = \frac{3\sqrt{3}}{2} \cdot \frac{J_3}{J_2^{3/2}} \quad (3-8)$$

The Lode angle is bounded to  $0 \leq \theta \leq \pi/3$  when the principal stresses are sorted as  $\sigma_{p1} \geq \sigma_{p2} \geq \sigma_{p3}$  (Ref. 1).

Following this convention,  $\theta = 0$  corresponds to the *tensile meridian*, and  $\theta = \pi/3$  corresponds to the *compressive meridian*. The Lode angle is part of a cylindrical

coordinate system (the Haigh–Westergaard coordinates) with height (hydrostatic axis)  $\xi = I_1/\sqrt{3}$  and radius  $r = \sqrt{2J_2}$ .



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 3-8](#).



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.

The octahedral plane (also called  $\pi$ -plane) is defined perpendicular to the hydrostatic axis in the Haigh–Westergaard coordinate system. The stress normal to this plane is  $\sigma_{\text{oct}} = I_1/3$ , and the shear stress on that plane is defined by

$$\tau_{\text{oct}} = \sqrt{2/3J_2}$$

The functions described in [Equation 3-6](#) and [Equation 3-7](#) 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.

The principal stresses ( $\sigma_{p1}$ ,  $\sigma_{p2}$ , and  $\sigma_{p3}$ ) can, when sorted as  $\sigma_{p1} \geq \sigma_{p2} \geq \sigma_{p3}$ , be written by using the invariants  $I_1$  and  $J_2$  and the Lode angle ([Ref. 1](#)):

$$\begin{aligned}\sigma_{p1} &= \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}} \cos \theta \\ \sigma_{p2} &= \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}} \cos \left( \theta - \frac{2\pi}{3} \right) \\ \sigma_{p3} &= \frac{1}{3}I_1 + \sqrt{\frac{4J_2}{3}} \cos \left( \theta + \frac{2\pi}{3} \right)\end{aligned}$$

### *Plane Strain and Plane Stress Cases*

For two-dimensional problems, there are two possible approximations: *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

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \quad (3-9)$$

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 3-9 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  $\frac{\partial w}{\partial z}$ .
- For anisotropic materials, two more degrees of freedom are added,  $uz$  and  $vz$ . They represent  $\frac{\partial u}{\partial z}$  and  $\frac{\partial v}{\partial z}$ .

### *Initial Stresses and Strains*

Initial stresses and strains refer to a stress and strain state that would exist even without the 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  $\epsilon_0$  and an initial stress  $s_0$  can be written as

$$s = s_0 + C : (\epsilon - \epsilon_0)$$

It can also be noted that the effect of the initial strain is analogous to that of a for example a thermal strain.

A common case is when you have results from another analysis or another physics interface, which you want to incorporate as initial stresses or strains. You should then use either the strain or the stress, but not both.

The **Initial Stress and Strain** node can be added to [Linear Elastic Material](#), [Nonlinear Elastic Materials](#), [Piezoelectric Material](#), [Elastoplastic Soil Models](#), [Magnetostrictive Material](#), or [Shape Memory Alloy](#).

Both the initial stress and strains 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.

#### **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 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.

#### *External Stress*

---

The external stress is a stress contribution which has a source other than the constitutive relation. It is similar to the initial stress described in the previous section, and the two features can to a large extent be used interchangeably.

There is however one large difference: using the External Stress, you can also prescribe a stress contribution that only acts as a load on the structure but is not added into the stress tensor. 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 is then

$$\delta W = \int_V -\delta \epsilon : \sigma_{\text{ext}} \, dv$$

The **External Stress** subnode can be added to [Linear Elastic Material](#), [Nonlinear Elastic Materials](#), [Elastoplastic Soil Models](#), [Shape Memory Alloy](#), or [Hyperelastic Material Models](#).

### IN SITU STRESS

The *vertical stress*,  $\sigma_v$ , 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,  $\rho$  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*  $\sigma_h$  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

$$\sigma_{\text{ins}} = \begin{bmatrix} \sigma_h & & \\ & \sigma_h & \\ & & \sigma_v \end{bmatrix}$$

### *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

$$I_1(\sigma) = \sum_{i,j} (\sigma^{ij} g_{ij}) = \sigma_r + \sigma_\phi + \sigma_z$$

# 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:

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \mathbf{F}_V - \nabla_X \cdot P^T \quad (3-10)$$

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 3-10 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

$$\begin{aligned} \rho_0 \frac{\partial^2 u_x}{\partial t^2} &= F_{V_x} - \left( \frac{\partial P_{xX}}{\partial X} + \frac{\partial P_{xY}}{\partial Y} + \frac{\partial P_{xZ}}{\partial Z} \right) \\ \rho_0 \frac{\partial^2 u_y}{\partial t^2} &= F_{V_y} - \left( \frac{\partial P_{yX}}{\partial X} + \frac{\partial P_{yY}}{\partial Y} + \frac{\partial P_{yZ}}{\partial Z} \right) \\ \rho_0 \frac{\partial^2 u_z}{\partial t^2} &= F_{V_z} - \left( \frac{\partial P_{zX}}{\partial X} + \frac{\partial P_{zY}}{\partial Y} + \frac{\partial P_{zZ}}{\partial Z} \right) \end{aligned}$$

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. Such tensors are called two-point tensors.

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:

$$\mathbf{F}_A = P \cdot \mathbf{n}_0$$

where the normal  $\mathbf{n}_0$  corresponds to the undeformed surface element. Such a force vector is often referred to as the *nominal traction*. 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,  $\sigma$ , 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 3-10 can be rewritten in terms of the Cauchy stress as

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \mathbf{f}_V - \nabla_x \cdot \sigma$$

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. The deviatoric part is defined as

$$\sigma_d = \sigma + p\mathbf{I}$$

The second invariant of the deviatoric stress

$$J_2(\sigma) = \frac{1}{2}\sigma_d:\sigma_d$$



is used for the computation of von Mises (equivalent) stress

$$\sigma_{\text{mises}} = \sqrt{3J_2(\boldsymbol{\sigma})}$$

# Material Models

There are many material models available for structural analysis in COMSOL Multiphysics. In this section, 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
  - Hyperbolic Law
  - Hardin–Drnevich
  - Duncan–Chang
  - Duncan–Selig
  - User Defined
- Linear Viscoelastic Materials
  - The Maxwell model
  - The Generalized Maxwell Model
  - Standard Linear Solid Model
  - The Kelvin–Voigt Model
  - The Generalized Kelvin–Voigt Model
  - The Burgers Model
  - Generalized Maxwell Model with Fractional Derivatives
  - Generalized Kelvin–Voigt model with fractional derivatives
  - Standard Linear Solid Model with Fractional Derivatives

- Hyperelastic Material Models
  - Neo-Hookean
  - St Venant–Kirchhoff
  - Mooney–Rivlin, Two Parameters
  - Mooney–Rivlin, Five Parameters
  - Mooney–Rivlin, Nine Parameters
  - Yeoh
  - Ogden
  - Storakers
  - Varga
  - Arruda–Boyce
  - Gent
  - Blatz–Ko
  - Gao
  - Murnaghan
  - Delfino
  - Fung
  - User Defined
  - Mullins Effect
- Elastoplastic Material Models
  - The von Mises Criterion
  - The Tresca Criterion
  - Hill Orthotropic Plasticity
- Porous Plasticity
  - The Shima–Oyane Criterion
  - The Gurson Criterion
  - The Gurson–Tvergaard–Needleman Criterion
  - The Fleck–Kuhn–McMeeking Criterion
  - The FKM–GTN Criterion
  - Capped Drucker–Prager

- Soil Plasticity
  - The Mohr–Coulomb Criterion
  - The Drucker–Prager Criterion
  - The Matsuoka–Nakai Criterion
  - The Lade–Duncan Criterion
- Failure Criteria for Concrete, Rocks, and Other Brittle Materials
  - The Bresler–Pister Criterion
  - The Willam–Warnke Criterion
  - The Ottosen Criterion
  - The Original Hoek–Brown Criterion
  - The Generalized Hoek–Brown Criterion
- Elastoplastic Soil Models
  - The Modified Cam-Clay Soil Model
  - The Modified Structured Cam-Clay Soil Model
  - The Extended Barcelona Basic Soil Model
  - The Hardening Soil Model
- Creep and Viscoplasticity
  - Norton Law (Power law)
  - The Norton–Bailey Law
  - Garofalo Law (Hyperbolic Sine Law)
  - Nabarro–Herring Creep (Diffusional Creep)
  - Coble Creep (Diffusional Creep)
  - Weertman Creep (Dislocation Creep)
  - Anand Viscoplastic Model
  - Chaboche Viscoplastic Model
  - Perzyna Viscoplastic Model
  - Creep Potential
  - Volumetric Creep
  - Deviatoric Creep
  - User Defined Creep
- Piezoelectric Material

- [Magnetostrictive Material](#)
- [Electrostriction](#)
- [Ferroelectroelasticity](#)
- [Rigid Domain Model](#)
- [Damage Models](#)
- [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:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\text{ex}} + \mathbf{C} : \boldsymbol{\varepsilon}_{\text{cl}} = \boldsymbol{\sigma}_{\text{ex}} + \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{inel}}) \quad (3-11)$$

where  $\mathbf{C}$  is the 4th order *elasticity tensor*, “:” stands for the double-dot tensor product (or double contraction). The elastic strain  $\boldsymbol{\varepsilon}_{\text{cl}}$  is the difference between the total strain  $\boldsymbol{\varepsilon}$  and all inelastic strains  $\boldsymbol{\varepsilon}_{\text{inel}}$ . There may also be an extra stress contribution  $\boldsymbol{\sigma}_{\text{ex}}$  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

$$W_{\text{s}} = \frac{1}{2} \boldsymbol{\varepsilon}_{\text{cl}} : (\mathbf{C} : \boldsymbol{\varepsilon}_{\text{cl}} + 2\boldsymbol{\sigma}_0) = \frac{1}{2} \boldsymbol{\varepsilon}_{\text{cl}} : (\boldsymbol{\sigma} + \boldsymbol{\sigma}_0) \quad (3-12)$$

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:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\ D_{12} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{13} & D_{23} & D_{33} & D_{34} & D_{35} & D_{36} \\ D_{14} & D_{24} & D_{34} & D_{44} & D_{45} & D_{46} \\ D_{15} & D_{25} & D_{35} & D_{45} & D_{55} & D_{56} \\ D_{16} & D_{26} & D_{36} & D_{46} & D_{56} & D_{66} \end{bmatrix} = \begin{bmatrix} C^{1111} & C^{1122} & C^{1133} & C^{1112} & C^{1123} & C^{1113} \\ C^{1122} & C^{2222} & C^{2233} & C^{2212} & C^{2223} & C^{2213} \\ C^{1133} & C^{2233} & C^{3333} & C^{3312} & C^{3323} & C^{3313} \\ C^{1112} & C^{2212} & C^{3312} & C^{1212} & C^{1223} & C^{1213} \\ C^{1123} & C^{2223} & C^{3323} & C^{1223} & C^{2323} & C^{2313} \\ C^{1113} & C^{2213} & C^{3313} & C^{1213} & C^{2313} & C^{1313} \end{bmatrix}$$

which is the *elasticity matrix*.

### ISOTROPIC MATERIAL AND ELASTIC MODULI

In this case, the elasticity matrix becomes

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (3-13)$$

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	$\nu =$	$\nu$	$\frac{E}{2G} - 1$	$\frac{13K-2G}{2 \cdot 3K+G}$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	$K =$	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	$K$	$\lambda + \frac{2\mu}{3}$
Shear modulus	$G =$	$\frac{E}{2(1+\nu)}$	$G$	$G$	$\mu$
Lamé parameter $\lambda$	$\lambda =$	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K - \frac{2G}{3}$	$\lambda$
Lamé parameter $\mu$	$\mu =$	$\frac{E}{2(1+\nu)}$	$G$	$G$	$\mu$
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  $\lambda$  and  $\mu$ ,

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (3-14)$$

or in terms of the bulk modulus  $K$  and shear modulus  $G$ :

$$D = \begin{bmatrix} K + \frac{4G}{3} & K - \frac{2G}{3} & K - \frac{2G}{3} & 0 & 0 & 0 \\ K - \frac{2G}{3} & K + \frac{4G}{3} & K - \frac{2G}{3} & 0 & 0 & 0 \\ K - \frac{2G}{3} & K - \frac{2G}{3} & K + \frac{4G}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad (3-15)$$

### 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:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ D_{12} & D_{22} & D_{23} & 0 & 0 & 0 \\ D_{13} & D_{23} & D_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & D_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{bmatrix} \quad (3-16)$$

where the components are as follows:

$$D_{11} = \frac{E_x^2(E_z v_{yz}^2 - E_y)}{D_{\text{denom}}}, \quad D_{12} = -\frac{E_x E_y (E_z v_{yz} v_{xz} + E_y v_{xy})}{D_{\text{denom}}}$$

$$D_{13} = -\frac{E_x E_y E_z (v_{xy} v_{yz} + v_{xz})}{D_{\text{denom}}}, \quad D_{22} = \frac{E_y^2 (E_z v_{xz}^2 - E_x)}{D_{\text{denom}}}$$

$$D_{23} = -\frac{E_y E_z (E_y v_{xy} v_{xz} + E_x v_{yz})}{D_{\text{denom}}}, \quad D_{33} = \frac{E_y E_z (E_y v_{xy}^2 - E_x)}{D_{\text{denom}}}$$

$$D_{44} = G_{xy}, \quad D_{55} = G_{yz}, \quad \text{and} \quad D_{66} = G_{xz}$$

where

$$D_{\text{denom}} = E_y E_z v_{xz}^2 - E_x E_y + 2v_{xy} v_{yz} v_{xz} E_y E_z + E_x E_z v_{yz}^2 + E_y^2 v_{xy}^2$$

The values of  $E_x$ ,  $E_y$ ,  $E_z$ ,  $\nu_{xy}$ ,  $\nu_{yz}$ ,  $\nu_{xz}$ ,  $G_{xy}$ ,  $G_{yz}$ , and  $G_{xz}$  are supplied in designated fields in the physics interface. COMSOL Multiphysics deduces the remaining components —  $\nu_{yx}$ ,  $\nu_{zx}$ , and  $\nu_{zy}$  — 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  $\nu_{xy}$  and  $\nu_{yx}$  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:

$$D_{44} = G_{yz}, \quad D_{55} = G_{xz}, \quad \text{and} \quad D_{66} = G_{xy}$$

If a pair of elastic moduli is present in the material definition, the values of  $E_x$ ,  $E_y$ ,  $E_z$ ,  $\nu_{xy}$ ,  $\nu_{yz}$ ,  $\nu_{xz}$ ,  $G_{xy}$ ,  $G_{yz}$ , 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.

#### 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 3-13](#) to [Equation 3-15](#). 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  $E_i$ ,  $\nu_{ij}$ , and  $G_{ij}$  are present in the material definition, the components of the symmetric elasticity matrix  $D$  are computed using [Equation 3-16](#). Note that the resulting elasticity matrix will not be fully anisotropic in neither case.

### AXIAL SYMMETRY

For the linear elastic material, the stress components in coordinate system are

$$\sigma^{ij} = \sigma_{ex}^{ij} + C^{ijkl} (\varepsilon_{kl} - \varepsilon_{inel,kl})$$

For anisotropic and orthotropic materials, the 4th-order elasticity tensor is defined from  $D$  matrix according to:

$$\begin{bmatrix} \sigma_r \\ \sigma_\varphi \\ \sigma_z \\ \sigma_{rz} \end{bmatrix} = \begin{bmatrix} \sigma_r \\ \sigma_\varphi \\ \sigma_z \\ \sigma_{rz} \end{bmatrix}_{ex} + D \left( \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_z \\ 2\varepsilon_{rz} \end{bmatrix} - \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_z \\ 2\varepsilon_{rz} \end{bmatrix}_{inel} \right)$$

The user input  $D$  matrix always contains the physical components of the elasticity tensor

$$C_{ijkl}^{phys}$$

and the corresponding tensor components are computed internally according to:

$$C^{ijkl} = \frac{C_{ijkl}^{phys}}{\sqrt{g_{ii}}\sqrt{g_{jj}}\sqrt{g_{kk}}\sqrt{g_{ll}}}$$

For an isotropic material:

$$C^{ijkl} = \lambda g^{ij} g^{kl} + \mu (g^{ik} g^{jl} + g^{il} g^{jk})$$

where  $\lambda$  and  $\mu$  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

$$S : \text{test}(\boldsymbol{\varepsilon}) = \sum_{i,j} S^{ij} \text{test}(\varepsilon_{ij})$$

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

$$F = \rho f_0(T) + W_s(\boldsymbol{\varepsilon}, T)$$

where the strain energy density  $W_s(\boldsymbol{\varepsilon}, T)$  is given by [Equation 3-12](#). Hence, the stress can be found as

$$\boldsymbol{\sigma} = \left( \frac{\partial F}{\partial \boldsymbol{\varepsilon}} \right)_T = \left( \frac{\partial W}{\partial \boldsymbol{\varepsilon}} \right)_T = \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{incl}})$$

and the entropy per unit volume can be calculated as

$$-\left( \frac{\partial F}{\partial T} \right)_\boldsymbol{\varepsilon} = \rho C_p \log(T/T_0) + S_{\text{elast}}$$

where  $T_0$  is a reference temperature, the volumetric heat capacity  $\rho C_p$  can be assumed to be independent of the temperature (Dulong–Petit law), and the elastic entropy is

$$S_{\text{elast}} = \boldsymbol{\alpha} : \boldsymbol{\sigma}$$

where  $\boldsymbol{\alpha}$  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

$$\rho C_p \frac{\partial T}{\partial t} + T \frac{\partial}{\partial t} S_{\text{elast}} = \nabla \cdot (k \nabla T) + Q_h$$

where  $k$  is the thermal conductivity matrix, and the heat source caused by the dissipation is

$$Q_h = \tau : \dot{\epsilon}$$

where  $\dot{\epsilon}$  is the strain-rate tensor and the tensor  $\tau$  represents all possible inelastic stresses (for example, a viscous stress).

Using the tensor components, the heat balance can be rewritten as:

$$\rho C_p \frac{\partial T}{\partial t} + \sum_{m,n} T \alpha_{mn} \frac{\partial}{\partial t} \sigma_{mn} = \nabla \cdot (k \nabla T) + Q_h \quad (3-17)$$

In many cases, the second term can be neglected in the left-hand side of Equation 3-17 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:

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot (\mathbf{C} : \nabla \mathbf{u})$$

where  $\mathbf{C}$  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:

$$\rho c^2 \mathbf{u}_n = \Gamma_n \cdot \mathbf{u}_n \quad (3-18)$$

where the Christoffel's tensor is defined as

$$\Gamma_n = \mathbf{n} \cdot \mathbf{C} \cdot \mathbf{n} \cdot$$

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 treated 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.

When the **Strain formulation** is selected in the **Use mixed formulation** list, the volumetric strain  $\epsilon_w$  is treated as an additional dependent variable.

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.



The mixed formulation is useful not only for linear elastic materials but also for elastoplastic materials, hyperelastic materials, and viscoelastic materials.

---



The order of the shape function for the auxiliary pressure variable should be one order less than that for the displacements. Thus, it is not recommended to use linear elements for the displacement variables in domains where the mixed formulation is turned on. Also 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

$$\frac{p_w - p}{K} = 0 \quad (3-19)$$

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  $\epsilon_w$  is used instead of the auxiliary pressure  $p_w$ , and it is the set equal to the volumetric strain  $\epsilon_{\text{vol}}$  using the equation

$$-K(\epsilon_w - \epsilon_{\text{vol}}) = 0 \quad (3-20)$$

The modified stress tensor then reads

$$\tilde{s} = s + K(\epsilon_w - \epsilon_{\text{vol}})\mathbf{I} = \text{dev}(s) + K\epsilon_w \mathbf{I}$$

The advantage of using the **Strain formulation** is that the values for the auxiliary strain  $\epsilon_w$  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 without geometric nonlinearity (and also for hyperelastic materials), the stress tensor  $s$  in the above equations is replaced by the 2nd Piola–Kirchhoff stress tensor  $S$ , see [Nearly Incompressible Hyperelastic Materials](#).



## *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:

$$\sigma = \sigma_{\text{ex}} + \mathbf{C} : \varepsilon_{\text{cl}} = \sigma_{\text{ex}} + \mathbf{C} : (\varepsilon - \varepsilon_{\text{incl}})$$

For a more detailed discussion, see [Equation 3-11](#).

It is possible to split the stress and elastic strain tensors into the deviatoric and volumetric contributions

$$\sigma = \text{dev}(\sigma) + \frac{1}{3}\text{trace}(\sigma)\mathbf{I}$$

and

$$\varepsilon_{\text{cl}} = \text{dev}(\varepsilon_{\text{cl}}) + \frac{1}{3}\text{trace}(\varepsilon_{\text{cl}})\mathbf{I}$$

Assuming only elastic stresses in linear isotropic elastic medium, Hooke’s law simplifies to

$$\sigma = \text{dev}(\sigma) + \frac{1}{3}\text{trace}(\sigma)\mathbf{I} = 2\mathbf{G}\text{dev}(\varepsilon_{\text{cl}}) + \mathbf{K}\text{trace}(\varepsilon_{\text{cl}})\mathbf{I}$$

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,

$$p = -\frac{1}{3}\text{trace}(\boldsymbol{\sigma})$$

The volumetric strain (positive in tension) is

$$\varepsilon_{\text{el,vol}} = \text{trace}(\boldsymbol{\varepsilon}_{\text{el}})$$

The linear relation between pressure and volumetric elastic strain is thus

$$p = -K\varepsilon_{\text{el,vol}}$$

The deviatoric stress and deviatoric elastic strain tensors are related by the shear modulus

$$\text{dev}(\boldsymbol{\sigma}) = 2G\text{dev}(\boldsymbol{\varepsilon}_{\text{el}})$$

By using the contraction of the deviatoric stress and strain tensors, we can alternatively relate the invariants of these tensors

$$J_2(\boldsymbol{\sigma}) = \frac{1}{2}\text{dev}(\boldsymbol{\sigma}):\text{dev}(\boldsymbol{\sigma}) = (2G)^2 J_2(\boldsymbol{\varepsilon}_{\text{el}})$$

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  $\varepsilon_{12} = \varepsilon_{21} = \gamma_{\text{el}}/2$ .

We can then write

$$J_2(\boldsymbol{\sigma}) = \frac{1}{2}\text{dev}(\boldsymbol{\sigma}):\text{dev}(\boldsymbol{\sigma}) = \tau^2$$

and

$$J_2(\boldsymbol{\varepsilon}_{\text{el}}) = \frac{1}{2}\text{dev}(\boldsymbol{\varepsilon}_{\text{el}}):\text{dev}(\boldsymbol{\varepsilon}_{\text{el}}) = \frac{1}{4}\gamma_{\text{el}}^2$$

The shear stress on the plane  $\tau = \sqrt{J_2(\boldsymbol{\sigma})}$  is then related to the elastic shear strain  $\gamma_{\text{el}} = 2\varepsilon_{12} = 2\sqrt{J_2(\boldsymbol{\varepsilon}_{\text{el}})}$  by the shear modulus

$$\tau = G\gamma_{\text{el}}$$

### *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,  $\tau = \sqrt{J_2(\boldsymbol{\sigma})}$  and  $\gamma_{el} = 2\sqrt{J_2(\boldsymbol{\varepsilon}_{el})}$  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. 1) is a nonlinear elastic material commonly used to model plastic deformation in metals, but it is 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

$$\varepsilon = \frac{\sigma}{E} + \varepsilon_{\text{ref}} \left( \frac{\sigma}{\sigma_{\text{ref}}} \right)^n$$

Here,  $E$  means the initial Young's modulus, and  $\varepsilon_{\text{ref}}$  is the strain at a reference stress  $\sigma_{\text{ref}}$ . The parameter  $n$  is the stress exponent. It is common to use  $\varepsilon_{\text{ref}} = 0.002$ , so  $\sigma_{\text{ref}}$  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,  $\sigma_{0.2} = 600$  GPa, and  $n = 4.8$ .

The linear strain is given by

$$\varepsilon_{\text{el}} = \frac{\sigma}{E}$$

and the nonlinear strain by

$$\varepsilon_{\text{nl}} = \varepsilon_{\text{ref}} \left( \frac{\sigma}{\sigma_{\text{ref}}} \right)^n$$

The total strain is the sum of linear and nonlinear strains

$$\varepsilon = \varepsilon_{\text{el}} + \varepsilon_{\text{nl}} = \frac{\sigma}{E} + \varepsilon_{\text{ref}} \left( \frac{\sigma}{\sigma_{\text{ref}}} \right)^n$$

In order to avoid a circular dependence of internal variables, the nonlinear strain  $\varepsilon_{\text{nl}}$  is defined with an auxiliary degree of freedom, so the stress reads  $\sigma = E(\varepsilon - \varepsilon_{\text{nl}})$ .

#### *Ramberg–Osgood Material in Soil Engineering*

In soil engineering, it is common to write the Ramberg–Osgood material with the stress-strain expression

$$\varepsilon = \frac{\sigma}{E} + \alpha \frac{\sigma_{\text{ref}}}{E} \left( \frac{\sigma}{\sigma_{\text{ref}}} \right)^n \quad (3-21)$$

so at the reference stress  $\sigma_{\text{ref}}$ , the strain is  $\varepsilon = (1 + \alpha)\sigma_{\text{ref}}/E$ . It is common to use  $\alpha = 3/7$ , so  $\sigma_{\text{ref}}$  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  $\varepsilon = \sigma_{\text{ref}}/E_{0.7}$ .

### POWER LAW

For this type of material, the shear stress is related to the elastic shear strain  $\gamma$  by the strain exponent  $n$  and a reference shear strain  $\gamma_{\text{ref}}$  (Ref. 2)

$$\tau = G_0 \gamma_{\text{ref}} \left( \frac{\gamma}{\gamma_{\text{ref}}} \right)^n$$

The secant shear modulus is given by the power law relation

$$G_s = G_0 \left( \frac{\gamma}{\gamma_{\text{ref}}} \right)^{n-1}$$

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:

$$K = K_t \text{ for } \varepsilon_{\text{el,vol}} > 0$$

and

$$K = K_c \text{ for } \varepsilon_{\text{el,vol}} < 0$$

where  $\varepsilon_{\text{el,vol}}$  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,  $\epsilon_{ax} = \epsilon_1$ , and the other two (transverse) strains are negative and related by the Poisson's ratio  $\epsilon_2 = \epsilon_3 = -\nu\epsilon_1$ .

For uniaxial compression, the axial strain is negative, and when the principal strains are sorted as  $\epsilon_1 > \epsilon_2 > \epsilon_3$  it corresponds to the third principal strain,  $\epsilon_{ax} = \epsilon_3$ . The other two (transverse) strains are positive and related by the Poisson's ratio  $\epsilon_1 = \epsilon_2 = -\nu\epsilon_3$ . 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

$$\epsilon_{vol,el} = \text{trace}(\epsilon_{cl})$$

and the elastic shear strain,

$$\gamma_{cl} = 2\sqrt{J_2(\epsilon_{cl})} = \sqrt{2\text{dev}(\epsilon_{cl}):\text{dev}(\epsilon_{cl})}.$$

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

$$\sigma_{ax} = E_s \epsilon_{ax}$$

Here,  $E_s$  is the secant Young's modulus, and the axial stress  $\sigma_{ax}$  is considered as a function of the elastic axial strain  $\epsilon_{ax}$ . Thus

$$E_s = \frac{\sigma_{ax}}{\epsilon_{ax}}$$

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  $\gamma$  and Poisson's ratio  $\nu$  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

$$\sigma_{ax} = \sigma_{ax}(\epsilon_{ax})$$

provided that

$$\sigma_{ax} = 0 \text{ when } \epsilon_{ax} = 0$$



The elastic axial strain  $\epsilon_{ax}$  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

$$\epsilon_{vol,el} = \text{trace}(\epsilon_{cl})$$

and the elastic shear strain,

$$\gamma = 2\sqrt{J_2(\epsilon_{cl})} = \sqrt{2\text{dev}(\epsilon_{cl}):\text{dev}(\epsilon_{cl})}.$$

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  $\tau$  and the elastic shear strain  $\gamma$  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

$$\sigma = 2G_s \text{dev}(\epsilon_{cl}) + K \text{trace}(\epsilon_{cl})I$$

where  $K$  is the bulk modulus.



The elastic shear strain  $\gamma$  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

$$G_s = G \frac{1}{1 + \left(\frac{\gamma}{\gamma_{\text{ref}}}\right)^n}$$

where the strain exponent  $n$  and a reference shear strain  $\gamma_{\text{ref}}$  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  $G_{\text{max}}$ . Sometimes it is also called “small strain shear modulus”.

### HARDIN–DRNEVICH

The Hardin–Drnevich model ([Ref. 3](#)) is a hyperbolic soil model (with  $n = 1$ ) defined by two input parameters: the initial shear modulus  $G$  and a reference shear strain  $\gamma_{\text{ref}}$

$$G_s = G \frac{1}{1 + \frac{\gamma}{\gamma_{\text{ref}}}}$$

This nonlinear soil model is commonly used for modeling soil dynamics in earthquake engineering problems.

Since  $\tau = G_s \gamma$ , the shear stress is bounded by  $\tau_{\text{max}} = G \gamma_{\text{ref}}$  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  $\gamma_{\text{ref}}$  is replaced by the reference shear strain at which the secant shear modulus has been decreased to 70% 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

$$G_s = G \frac{1}{1 + \frac{3}{7} \frac{\gamma}{\gamma_{0.7}}}$$

so that when  $\gamma = \gamma_{0.7}$  the secant shear modulus is  $G_s = 0.7G$ .

### **DUNCAN–CHANG**

The original model was originated by Kondner to fit triaxial test data for undrained soils. Duncan and Chang (Ref. 4) and other coworkers (Ref. 5) developed this hyperbolic model to its current state. The material model is written in terms of the axial and radial stresses  $\sigma_1$  and  $\sigma_3$  and the axial strain  $\epsilon$ , and it describes the stress-strain curve by fitting the hyperbola

$$\sigma_1 - \sigma_3 = \frac{\epsilon}{a + b\epsilon}$$

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  $q_{\text{ult}}$

$$\frac{1}{b} = (\sigma_1 - \sigma_3)_{\text{ult}} = q_{\text{ult}}$$

The ultimate value  $q_{\text{ult}}$  is related to the strength of the soil.

For the triaxial test, the axial strain  $\varepsilon$  is related to the shear strain  $\gamma$  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

$$\tau = \frac{\frac{1}{2(1+\nu)}\gamma}{\frac{1}{E} + \frac{1}{q_{\text{ult}}}\frac{\sqrt{3}}{2(1+\nu)}\gamma}$$

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

$$\tau = \frac{\gamma}{\frac{1}{G} + \frac{\sqrt{3}\gamma}{q_{\text{ult}}}}$$

which is an hyperbolic law with a secant shear modulus of

$$G_s = \frac{G}{1 + \frac{G}{q_{\text{ult}}}\sqrt{3}\gamma}$$

### **DUNCAN–SELIG**

The Duncan–Selig model is a combination of the Duncan's hyperbolic material model (Ref. 4, Ref. 5) and Selig's model to describe nonlinear bulk modulus behavior. Selig (Ref. 6) 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

$$p = -K \frac{\varepsilon_{\text{el,vol}}}{1 - \frac{\varepsilon_{\text{el,vol}}}{\varepsilon_{\text{ult}}}}$$

where  $\epsilon_{el,vol}$  is the volumetric strain, and  $\epsilon_{ult}$  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

$$K_s = K \frac{1}{1 - \frac{\epsilon_{el,vol}}{\epsilon_{ult}}}$$

### 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  $\epsilon_{vol,el}$  can be called in user defined expressions by referencing the variables `solid.eelvol`, where `solid` is the name of the physics interface node.

### Linear Viscoelastic Materials

*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.

### THE GENERALIZED MAXWELL MODEL

For isotropic linear elastic materials in the absence of inelastic stresses, Hooke's law in [Equation 3-11](#) reduces to

$$\sigma = C : \epsilon_{el}$$

where the elastic strain tensor  $\epsilon_{el} = \epsilon - \epsilon_{inel}$  represents the total strain minus initial and inelastic strains, such as thermal strains.

The stress tensor can be decomposed into a pressure and a deviatoric stress:

$$\sigma = -pI + \sigma_d$$

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

$$\sigma_d = \text{dev}(\sigma) = \sigma + pI$$

The elastic strain tensor  $\varepsilon_{el}$  can in the same way be decomposed into volumetric and deviatoric components

$$\varepsilon_{el} = \frac{1}{3}\varepsilon_{el,vol}I + \varepsilon_d$$

with the volumetric elastic strain given by

$$\varepsilon_{el,vol} = \text{trace}(\varepsilon_{el})$$

and the deviatoric contribution by

$$\varepsilon_d = \text{dev}(\varepsilon_{el})$$

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  $G$

$$\sigma_d = \text{dev}(\sigma) = 2G\text{dev}(\varepsilon_{el}) = 2G\varepsilon_d$$

The total stress in [Equation 3-11](#) is then

$$\sigma = K\varepsilon_{el,vol}I + 2G\varepsilon_d$$

In case of geometric nonlinearity,  $\sigma$  represents the second Piola–Kirchhoff stress tensor and  $\varepsilon_{el}$  the elastic Green–Lagrange strain tensor.

For viscoelastic materials, the deviatoric stress  $\sigma_d$  is not linearly related to the deviatoric strain  $\epsilon_d$  but it also depends on the strain history. It is normally defined by the hereditary integral:

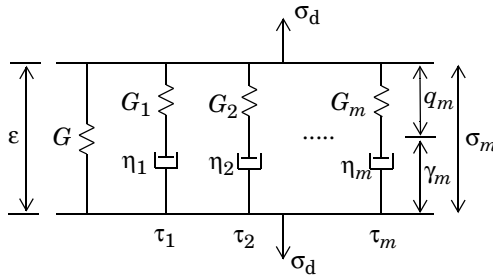
$$\sigma_d = 2 \int_0^t \Gamma(t-t') \frac{\partial \epsilon_d}{\partial t'} dt'$$

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  $\tau_m$  is the *relaxation time* constant of the spring-dashpot pair in branch  $m$ .

The relaxation time  $\tau_m$  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 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  $\gamma_m = \varepsilon - q_m$  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  $G_\infty$ . 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 the 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 [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 on any form, for example in terms of Young's modulus,  $E$ , and Poisson's ratio,  $\nu$ . 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  $\gamma_m$

$$\sigma_m = 2G_m q_m = 2\eta_m \dot{\gamma}_m = 2G_m(\varepsilon - \gamma_m)$$

The sum of the stresses in the viscoelastic branches is then computed from

$$\sigma_q = \sum_{m=1}^N \sigma_m = \sum_{m=1}^N 2G_m(\varepsilon - \gamma_m)$$

The total stress in Hooke's law ([Equation 3-11](#)) is then augmented by the viscoelastic stress  $\sigma_q$

$$\sigma = \sigma_0 + \mathbf{C} : \varepsilon_{el} + \sigma_q \quad (3-22)$$

#### Computing the Stress on Each Branch

The auxiliary variable  $\gamma_m$  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

$$\sigma_m = 2G_m q_m = 2\eta_m \dot{\gamma}_m = 2G_m(\varepsilon - \gamma_m)$$



the auxiliary variables  $\gamma_m$  can be computed by solving the ODE

$$\eta_m \dot{\gamma}_m = G_m (\varepsilon - \gamma_m) \quad (3-23)$$

The relation between viscosity and relaxation time is

$$\eta_m = G_m \tau_m$$

so that [Equation 3-23](#) can equivalently be written as

$$\tau_m \dot{\gamma}_m + \gamma_m = \varepsilon \quad (3-24)$$

The viscoelastic strain variables  $\gamma_m$  are treated as additional degrees of freedom. The shape functions are chosen to be one order lower than those used for the displacements because these variables add to the strains and stresses computed from displacement derivatives. Alternatively, [Equation 3-37](#) can be solved using a [Local Time Integration](#) algorithm.



The viscoelastic strain variables  $\gamma_m$  are called `solid.lemm1.vis1.ev`, where `solid` is the name of the physics interface node, and `lemm1` is the name of the elastic material node.

### Frequency Domain

In 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

$$G_{GM} = G + \sum_{m=1}^N G_m \frac{j\omega\tau_m}{1+j\omega\tau_m}$$

The storage and loss moduli are then computed as the real and imaginary parts of the complex shear modulus

$$G' = \text{real}(G_{GM}) = G + \sum_{m=1}^N G_m \frac{(\omega\tau_m)^2}{1 + (\omega\tau_m)^2}$$

and

$$G'' = \text{imag}(G_{GM}) = \sum_{m=1}^N G_m \frac{\omega\tau_m}{1 + (\omega\tau_m)^2}$$

### Energy Dissipation

The *dissipated energy density rate* (SI unit: W/m<sup>3</sup>) 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

$$\dot{W}_v = \sum_{m=1}^N \dot{W}_m$$

In order to compute the dissipated energy density, the variable  $\dot{W}_v$  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

$$W_v = \omega G'' \tilde{\epsilon}_d : \text{conj}(\tilde{\epsilon}_d)$$

### THE MAXWELL MODEL

The *Maxwell* model is a simplification of the generalized Maxwell model with only one spring-dashpot branch:

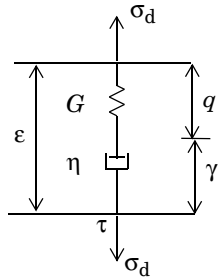


Figure 3-2: Maxwell model

The compliance for the Maxwell model is defined as the sum of the compliance modulus of the elastic branch plus the compliance in the damper

$$J_M = \frac{1}{G} + \frac{1}{j\omega\tau G}$$

and the complex shear modulus reads

$$G_M = \frac{1}{J_M} = \frac{j\omega\tau}{1+j\omega\tau}G$$

The storage and loss moduli are then computed as the real and imaginary parts of the complex shear modulus

$$G' = \text{real}(G_M) = \frac{(\omega\tau)^2}{1 + (\omega\tau)^2}G$$

and

$$G'' = \text{imag}(G_M) = \frac{\omega\tau}{1 + (\omega\tau)^2}G$$

### THE GENERALIZED KELVIN-VOIGT MODEL

The *generalized Kelvin-Voigt* model is used to simulate 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  $\epsilon_d$  is not linearly related to the deviatoric stress  $\sigma_d$ , but it also depends on the strain history. It is normally defined by the hereditary integral:

$$\epsilon_d = \int_0^t \psi(t-t') \frac{\partial \sigma_d}{2\partial t'} dt'$$

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.

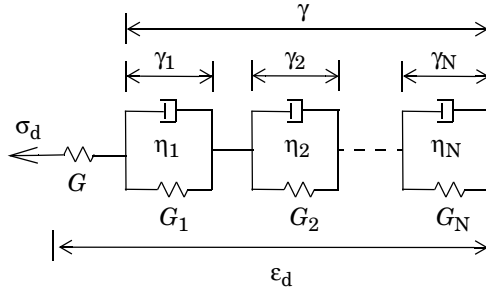


Figure 3-3: The generalized Kelvin–Voigt model.

Here,  $G$  is the stiffness of the main elastic branch,  $G_m$  represents the stiffness of the spring in branch  $m$ , and  $\tau_m$  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 relaxation time  $\tau_m$  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 of each branch can be expressed in terms of the shear modulus and relaxation time as

$$\eta_m = G_m \tau_m = \tau_m / J_m$$

The auxiliary strain variables  $\gamma_m$  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

$$\gamma = \sum_m \gamma_m \quad (3-25)$$

The deviatoric stress  $\sigma_d$  in the pure elastic branch is the same as in all the Kelvin–Voigt elements

$$\sigma_d = 2G \text{dev}(\varepsilon_{el}) = 2G_m \gamma_m + 2\eta_m \dot{\gamma}_m \quad (3-26)$$

here, the deviatoric elastic strain is defined by the difference between the total and inelastic strains.

#### Computing the Strain on Each Branch

The auxiliary strain variables  $\gamma_m$  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,  $\gamma_m$  is the strain in the element  $m$ , and  $\eta_m = \tau_m G_m$  is the viscosity. The auxiliary variables  $\gamma_m$  are computed by solving the ODE

$$\gamma_m + \tau_m \dot{\gamma}_m = \frac{1}{2G_m} \sigma_d \quad (3-27)$$

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  $J_\infty$  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 branch plus the complex compliances in the viscoelastic branches

$$J_{\text{GVK}} = \frac{1}{G} + \sum_{m=1}^N \frac{1}{G_m} \frac{1}{1 + j\omega\tau_m}$$

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<sup>3</sup>) 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

$$\dot{W}_v = \sum_{m=1}^N \dot{W}_m$$

In order to compute the dissipated energy density, the variable  $\dot{W}_v$  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

$$W_v = \omega G'' \tilde{\epsilon}_d : \text{conj}(\tilde{\epsilon}_d)$$

#### THE KELVIN–VOIGT MODEL

The *Kelvin–Voigt viscoelastic model* is represented by a spring connected in parallel with a dashpot:

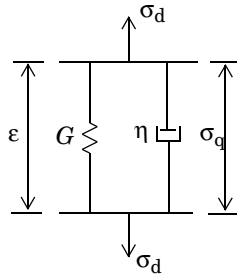


Figure 3-4: The Kelvin–Voigt model.

The stress tensor in the viscous branch is computed from the elastic strain rate

$$\sigma_q = 2\eta \dot{\epsilon} = 2G\tau \dot{\epsilon} \quad (3-28)$$

so there is no need to add the extra DOFs to compute the auxiliary strain tensor  $\gamma$ .

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

$$\dot{W}_v = \sigma_q : \dot{\varepsilon}$$



The Kelvin–Voigt model is equivalent to viscous damping, see [Viscous Damping Theory](#).

### 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:

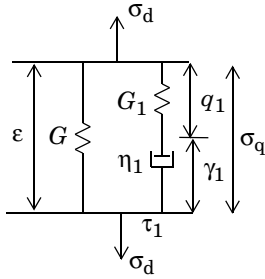


Figure 3-5: Standard linear solid (SLS) model.

The stress in the single branch is computed as

$$\sigma_q = 2G_1 q_1 = 2\eta_1 \dot{\gamma}_1 = 2G_1(\varepsilon - \gamma_1)$$

where the relaxation time is related to the stiffness and relaxation time as  $\eta_1 = \tau_1 G_1$ .

The auxiliary strain tensor  $\gamma_1$  is computed after solving by the ODE

$$\tau_1 \dot{\gamma}_1 + \gamma_1 = \varepsilon$$

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  $G_0 = G + G_1$ .



### 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](#):

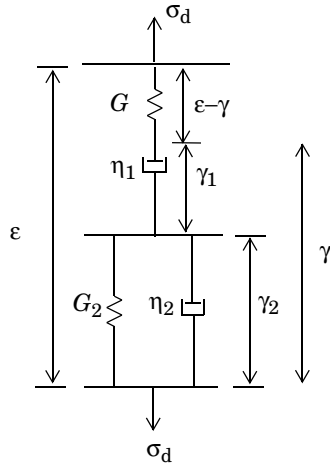


Figure 3-6: The Burgers model.

The strain in the first dashpot follows the ODE

$$\text{dev}(\dot{\sigma}) = 2G(\dot{\epsilon} - \dot{\gamma}) = 2\eta_1\dot{\gamma}_1$$

where the shear modulus  $G$  is taken from the parent Linear Elastic material.

The strain in the second dashpot follows the ODE

$$2\eta_2\dot{\gamma}_2 = \text{dev}(\dot{\sigma}) - 2G_2\dot{\gamma}_2$$

The total strain in the dashpots is computed from

$$\dot{\gamma} = \dot{\gamma}_1 + \dot{\gamma}_2$$

Combing these equations, it is possible to recover a second-order ODE for the strain tensor  $\gamma$ ,

$$\ddot{\gamma}_2 = \left( \frac{1}{2\eta_1} + \frac{1}{2\eta_2} \right) \text{dev}(\dot{\sigma}) - \frac{G_2}{\eta_2} \dot{\gamma}_2 + \frac{G_2}{2\eta_1\eta_2} \text{dev}(\sigma)$$

Note that the Burgers material has two relaxation times related to the stiffness and viscosity in the springs and dashpots. The relaxation times  $\tau_1$  and  $\tau_2$  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. 34). In some cases, viscoelasticity models with fractional derivatives have shown to better match experimental data.

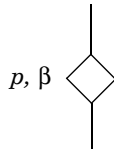


Figure 3-7: Spring-pot element.

The basic stress-strain relation of the spring-pot element is given by a proportionality factor  $p$  and a *fractional time derivative* of order  $\beta$

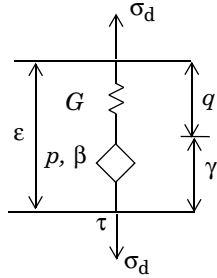
$$\text{dev}(\boldsymbol{\sigma}) = 2p \frac{d^\beta}{dt^\beta} \boldsymbol{\varepsilon} = 2p D^\beta \boldsymbol{\varepsilon}$$

The fractional order  $\beta$  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 stress strain relation is given by

$$\text{dev}(\boldsymbol{\sigma}) = 2p D^\beta \boldsymbol{\gamma} = 2G(\boldsymbol{\varepsilon} - \boldsymbol{\gamma})$$

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, we can substitute the fractional time derivative operator by

$$D^\beta \rightarrow (j\omega)^\beta$$



The fractional derivative models are available for frequency domain analyses only.

Thus, the stress-strain relationship reads

$$\text{dev}(\sigma) = 2 \frac{Gp(j\omega)^\beta}{G + p(j\omega)^\beta} \varepsilon$$

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  $\tau$  for a branch with a spring and a spring-pot element in series is derived from

$$\tau^\beta = \frac{p}{G}$$

so we can write  $p = \tau^\beta/G$ , and the stress-strain relationship in the spring-spring-pot system reads

$$\text{dev}(\sigma) = 2 \frac{G(j\omega\tau)^\beta}{1 + (j\omega\tau)^\beta} \varepsilon$$

The complex-valued shear modulus for this Maxwell element reads

$$G_M = \frac{(j\omega\tau)^\beta}{1 + (j\omega\tau)^\beta} G$$

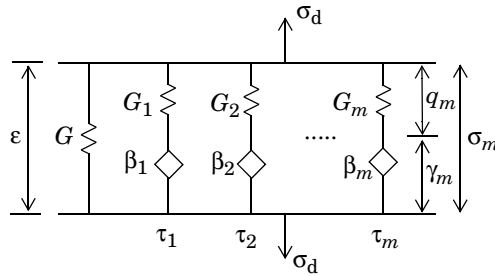
Storage and loss moduli are defined as the real and imaginary part of the complex shear modulus

$$G' = \text{real}(G_M) \text{ and } G'' = \text{imag}(G_M)$$

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  $G_m$ ,  $\tau_m$  and  $\beta_m$  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.

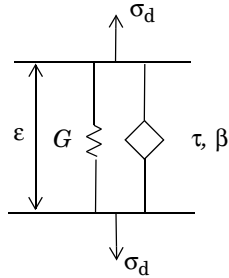
$$G' = \text{real}(G_{GM}) \text{ and } G'' = \text{imag}(G_{GM})$$

### 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

$$\text{dev}(\sigma) = 2pD^\beta \varepsilon + 2G\varepsilon$$

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, we can substitute the fractional time derivative operator by

$$D^\beta \rightarrow (j\omega)^\beta$$

The stress-strain relationship for this Kelvin element then reads

$$\text{dev}(\sigma) = 2(p(j\omega)^\beta + G)\varepsilon$$

The relaxation time  $\tau$  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

$$\text{dev}(\sigma) = 2G((j\omega\tau)^\beta + 1)\varepsilon$$

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.

$$G' = \text{real}(G_{KV}) = (1 + (\omega\tau)^\beta \cos \beta\pi/2)G$$

and

$$G'' = \text{imag}(G_{KV}) = (\omega\tau)^\beta \sin(\beta\pi/2)G$$

The loss factor is computed from

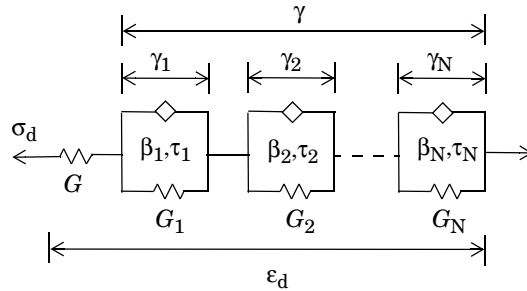
$$\eta_{KV} = \frac{G''}{G'} = \frac{(\omega\tau)^\beta \sin \beta\pi/2}{1 + (\omega\tau)^\beta \cos \beta\pi/2}$$

The complex-valued compliance of the *Kelvin-Voigt element* with fractional time derivative reads

$$J_{KV} = \frac{1}{G_{KV}} = \frac{1}{G} \frac{1}{1 + (j\omega\tau)^\beta}$$

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

$$J_{GKV} = \frac{1}{G} + \sum_m \frac{1}{1 + (j\omega\tau_m)^\beta} \frac{1}{G_m}$$

where  $G_m$ ,  $\tau_m$  and  $\beta_m$  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

$$G_{\text{GKV}} = \frac{1}{J_{\text{GKV}}}$$

$$G' = \text{real}(G_{\text{GKV}}) \text{ and } G'' = \text{imag}(G_{\text{GKV}})$$

### 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.

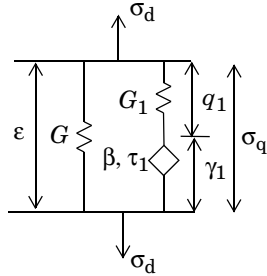


Figure 3-12: Standard linear solid (SLS) model with fractional derivatives.

The deviatoric stress in the spring-spring-pot branch is computed as

$$\sigma_q = 2G_1(\epsilon - \gamma_1) = 2pD^\beta \gamma_1$$

here,  $\gamma_1$  is the strain in the spring-pot element. In frequency domain this equation translates to

$$\sigma_q = 2p(j\omega)^\beta \gamma_1 = \frac{G_1}{G_1 + p(j\omega)^\beta} 2p(j\omega)^\beta \epsilon$$

Using the relaxation time  $\tau = (p/G_1)^{1/\beta}$ , this reads

$$\sigma_q = 2 \frac{G_1}{1 + (j\omega\tau)^\beta} (j\omega\tau)^\beta \epsilon = 2G_q \epsilon$$

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

$$G' = \text{real}(G_{SLS}) \text{ and } G'' = \text{imag}(G_{SLS})$$

### 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](#):

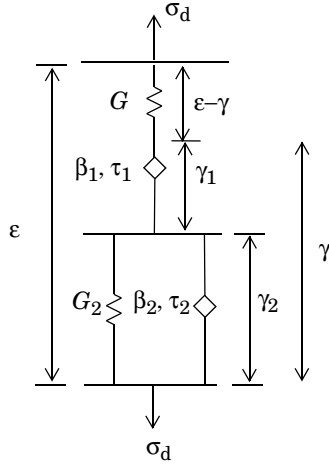


Figure 3-13: The Burgers model with fractional derivatives.

The deviatoric stress in the main branch is computed as

$$\sigma_d = 2G(\epsilon - \gamma) = 2p_1 D^{\beta_1} \gamma_1$$

here,  $\gamma_1$  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

$$\frac{\sigma_d}{2G} = \epsilon - \gamma = (j\omega\tau_1)^{\beta_1} \gamma_1$$

The stress in the second spring-pot follows



$$2p_2 D^{\beta_2} \gamma_2 = \sigma_d - 2G_2 \gamma_2$$

here,  $\gamma_2$  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

$$\gamma_1 = \frac{1}{2G(j\omega\tau_1)^{\beta_1}} \sigma_d \quad \text{and} \quad \gamma_2 = \frac{1}{2G_2(1 + (j\omega\tau_2)^{\beta_2})} \sigma_d$$

gives the stress and strain relation for Burgers model with fractional derivatives

$$\left( \frac{1}{2G} + \frac{1}{2G(j\omega\tau_1)^{\beta_1}} + \frac{1}{2G_2(1 + (j\omega\tau_2)^{\beta_2})} \right) \sigma_d = \varepsilon$$

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

$$G' = \text{real}(G_B) \quad \text{and} \quad G'' = \text{imag}(G_B)$$

## 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 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 3-24 for a TRS material is modified to

$$\alpha_T(T)\tau_m\dot{\gamma}_m + \gamma_m = \varepsilon$$

For the SLS model, the shift applies to a single branch

$$\alpha_T(T)\tau_1\dot{\gamma}_1 + \gamma_1 = \varepsilon$$

and for the Kelvin–Voigt model, it applies to the viscosity in the dashpot

#### Williams–Landel–Ferry Shift

One commonly used shift function is defined by the WLF (Williams–Landel–Ferry) equation:

$$\log(\alpha_T) = \frac{-C_1(T - T_0)}{C_2 + (T - T_0)}$$

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  $T_0$ . 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  $T_0$ , 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  $T - T_0$ .



The shift factor at the reference temperature equals  $\alpha_T(T_0) = 1$ , so that  $T_0$  is the temperature at which the master curve is given. If the temperature  $T$  drops below  $T_0 - C_2$ , the WLF equation is no longer valid.

Since the master curve is measured at an arbitrary reference temperature  $T_0$ , the shift factor  $\alpha_T(T)$  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

$$\log(\alpha_T) = \frac{Q}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right)$$

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

$$\log(\alpha_T) = \frac{Q}{R} \left( \frac{\chi}{T} + \frac{1-\chi}{T_f} - \frac{1}{T_0} \right)$$

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,  $T_0$  is a reference temperature,  $\chi$  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.

$$T_f = \sum_i w_i T_{fi} \quad \text{with} \quad \sum_i w_i = 1$$

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,  $\lambda_{0i}$  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 analysis, the frequency decomposition is performed as

$$\sigma_d = \text{real}(\tilde{s}_d e^{j\omega t})$$

$$\varepsilon_d = \text{real}(\tilde{\varepsilon}_d e^{j\omega t})$$

Equation 3-22 and Equation 3-24 are then simplified to

$$\tilde{\sigma}_d = 2(G' + jG'')\tilde{\varepsilon}_d$$

where the *shear storage modulus*  $G'$  and the *shear loss modulus*  $G''$  are defined for the generalized Maxwell model as

$$G' = G + \sum_{m=1}^N G_m \frac{(\omega\tau_m)^2}{1 + (\omega\tau_m)^2} \quad \text{and} \quad G'' = \sum_{m=1}^N G_m \frac{\omega\tau_m}{1 + (\omega\tau_m)^2}$$

for the SLS model as

$$G' = G + G_1 \frac{(\omega\tau_1)^2}{1 + (\omega\tau_1)^2} \quad \text{and} \quad G'' = G_1 \frac{\omega\tau_1}{1 + (\omega\tau_1)^2}$$

for the Kelvin–Voigt model as

$$G' = G \quad \text{and} \quad G'' = \eta\omega = G\tau\omega$$

The internal work of viscous forces averaged over a time period  $2\pi/\omega$  is computed as

$$Q_h = \omega G^n \tilde{\epsilon}_d : \text{conj}(\tilde{\epsilon}_d)$$

### LOCAL TIME INTEGRATION

For the [The Generalized Maxwell Model](#) and the [Standard Linear Solid Model](#) it is possible to use a local time integration algorithm. 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 originally suggested in [Ref. 18](#), and it is schematically described in the following.

Each branch in the Generalized Maxwell model is characterized by an ODE on the following form

$$\tau_m \dot{\gamma}_m + \gamma_m = \epsilon(t) \quad (3-29)$$

where  $\gamma_m$  is the viscoelastic strain in the branch. The exact solution to [Equation 3-29](#) is

$$\gamma_m^{n+1} = \exp\left(-\frac{\Delta t}{\tau_m}\right) \gamma_m^n + \frac{1}{\tau_m} \int_{t^n}^{t^{n+1}} \exp\left(-\frac{t^{n+1}-\tau}{\tau_m}\right) \epsilon(\tau) d\tau \quad (3-30)$$

By assuming that the strain  $\epsilon(t)$  varies linearly within each time increment, the integral in [Equation 3-30](#) is analytically computed, so the viscoelastic strain at increment  $n+1$  reads

$$\gamma_m^{n+1} = \epsilon^{n+1} - \exp\left(-\frac{\Delta t}{\tau_m}\right) (\epsilon^n - \gamma_m^n) - \frac{1 - \exp\left(-\frac{\Delta t}{\tau_m}\right)}{\frac{\Delta t}{\tau_m}} (\epsilon^{n+1} - \epsilon^n) \quad (3-31)$$

All variables in [Equation 3-31](#) evaluated at increment  $n$  are stored as internal state variables, and the equation can be applied to each branch of the Generalized Maxwell model. The same implementation is also used 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  $\infty$  to denote the long-term equilibrium (as  $t \rightarrow \infty$ ).

The second Piola–Kirchhoff stress is computed from

$$\mathbf{S} = 2 \frac{\partial W_s}{\partial \mathbf{C}} = \mathbf{S}_{\text{iso}} + \mathbf{S}_{\text{vol}} + \sum_{m=1}^N \mathbf{Q}_m$$

where the auxiliary second Piola–Kirchhoff stress tensors  $\mathbf{Q}_m$  are defined as

$$\mathbf{Q}_m = 2 \frac{\partial \Psi_m}{\partial \mathbf{C}}$$

The time evolution of the auxiliary stress tensor  $\mathbf{Q}_m$  in each viscoelastic branch is given by the rate

$$\dot{\mathbf{Q}}_m + \frac{1}{\tau_m} \mathbf{Q}_m = \dot{\mathbf{S}}_{\text{iso},m}$$

here,  $\mathbf{S}_{\text{iso},m}$  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*  $\beta_m$  as

$$\mathbf{S}_{\text{iso},m} = 2 \frac{\partial W_{\text{iso},m}}{\partial \mathbf{C}} = 2 \beta_m \frac{\partial W_{\text{iso}}}{\partial \mathbf{C}} = \beta_m \mathbf{S}_{\text{iso}}$$

so the time evolution of the auxiliary stress tensor  $\mathbf{Q}_m$  is given by

$$\dot{\mathbf{Q}}_m + \frac{1}{\tau_m} \mathbf{Q}_m = \beta_m \dot{\mathbf{S}}_{\text{iso}}$$

This equation is not well suited for modeling prestressed bodies. Applying the change of variables

$$q_m = \beta_m S_{\text{iso}} - Q_m$$

the time evolution of the auxiliary stress tensor  $q_m$  reads

$$\tau_m \dot{q}_m + q_m = \beta_m \dot{S}_{\text{iso}}$$

### TEMPERATURE EFFECTS

The same options for defining [Temperature Effects](#) as described for [Linear Viscoelastic Materials](#) are available for large strain viscoelasticity.

### *Hyperelastic Material Models*

---

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 in the local coordinate system 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.S111 = 2*d(solid.Ws,solid.C111)`, `solid.S112 = d(solid.Ws,solid.C112)` etc. 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](#)

### THEMAL EXPANSION

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  $\lambda_{th}$ , which for linear thermal expansion in isotropic materials can be written in terms of the isotropic coefficient of thermal expansion,  $\alpha_{iso}$ , and the absolute change in temperature

$$J_{th} = \lambda_{th}^3 \quad \text{and} \quad \lambda_{th} = 1 + \alpha_{iso}(T - T_{ref})$$

Here, the term  $\alpha_{iso}(T - T_{ref})$  is the *thermal strain*. The isotropic thermal gradient is therefore a diagonal tensor defined as

$$F_{th} = \lambda_{th}I$$

When the coefficient of thermal expansion  $\alpha$  is anisotropic, the thermal strain is computed from

$$\epsilon_{th} = \alpha(T - T_{ref})$$

and the anisotropic thermal gradient is defined as



$$F_{th} = I + \varepsilon_{th}$$



The internal variables for the thermal stretch and the thermal volume ratio are named `solid.stchth` and `solid.Jth`.

### HYGROSCOPIC SWELLING

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain depends linearly on the moisture content

$$\varepsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where  $\beta_h$  is the coefficient of hygroscopic swelling,  $c_{mo}$  is the moisture concentration, and  $c_{mo,ref}$  is the strain-free reference concentration. The coefficient of hygroscopic swelling can represent isotropic or anisotropic swelling. The anisotropic hygroscopic gradient is defined as

$$F_{hs} = I + \varepsilon_{hs}$$

### HYPERELASTICITY WITH PLASTICITY

It is possible to combine the hyperelastic material models with plasticity. Since these models are primarily used for large strain applications, only the large strain plasticity formulation is available. The decomposition between elastic and plastic deformation is made using a multiplicative decomposition of the deformation gradient tensor,

$$F_{el} = FF_{pl}^{-1}$$

Here, the plastic deformation tensor  $F_{pl}$  depends on the plastic flow rule, yield function, and plastic potential.



- [Multiplicative Decomposition](#)
- [Plastic Flow 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

$$\mathbf{F}_{\text{el}} = \mathbf{F}_{\text{el,vol}} \overline{\mathbf{F}}_{\text{el}}$$

with  $\mathbf{F}_{\text{el,vol}}$  as the volumetric elastic deformation (a diagonal tensor) and  $\overline{\mathbf{F}}_{\text{el}}$  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

$$J_{\text{el}} = \det(\mathbf{F}_{\text{el}}) = \det(\mathbf{F}_{\text{el,vol}})$$

and the volumetric deformation as

$$\mathbf{F}_{\text{el,vol}} = J_{\text{el}}^{1/3} \mathbf{I}$$

By using  $J_{\text{el}}$  it is possible to define the *isochoric-elastic deformation* gradient

$$\overline{\mathbf{F}}_{\text{el}} = J_{\text{el}}^{-1/3} \mathbf{F}_{\text{el}}$$

the *isochoric-elastic right Cauchy–Green* tensor

$$\overline{\mathbf{C}}_{\text{el}} = \overline{\mathbf{F}}_{\text{el}}^T \overline{\mathbf{F}}_{\text{el}} = J_{\text{el}}^{-2/3} \mathbf{C}_{\text{el}}$$

and the *isochoric-elastic Green–Lagrange strain* tensor

$$\overline{\boldsymbol{\varepsilon}}_{\text{el}} = \frac{1}{2}(\overline{\mathbf{C}}_{\text{el}} - \mathbf{I})$$

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  $\overline{\mathbf{F}}_{\text{el}}$  and  $\overline{\mathbf{C}}_{\text{el}}$  the *modified* tensors. Note that

$$\det(\overline{\mathbf{F}}_{\text{el}}) = \det(\overline{\mathbf{C}}_{\text{el}}) = 1$$



The internal variables for the isochoric-elastic Cauchy–Green deformation tensor in local coordinate system are named `solid.CIe111`, `solid.CIe112`, and so on.

---

The other two invariants normally used together with  $J_{\text{el}}$  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.I1C1e1`, and `solid.I2C1e1`.

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

$$I_1(\overline{\epsilon}_{el}) = \text{trace}(\overline{\epsilon}_{el}) = \frac{1}{2}(I_1(\overline{C}_{el}) - 3)$$

$$I_2(\overline{\epsilon}_{el}) = \frac{1}{2}(I_1^2(\overline{\epsilon}_{el}) - \text{trace}(\overline{\epsilon}_{el}^2)) = \frac{1}{4}(I_2(\overline{C}_{el}) - 2I_1(\overline{C}_{el}) + 3)$$

$$I_3(\overline{\epsilon}_{el}) = \det(\overline{\epsilon}_{el}) = \frac{1}{8}(I_1(\overline{C}_{el}) - I_2(\overline{C}_{el}))^2$$



The internal variables for the invariants of the isochoric elastic Green–Lagrange strain tensor are named `solid.I1e1e1`, `solid.I2e1e1`, and `solid.I3e1e1`.

### NEARLY INCOMPRESSIBLE HYPERELASTIC MATERIALS

If the **Nearly incompressible material** is selected in the **Compressibility** list in the [Hyperelastic Material](#) node, the total strain energy density  $W_s$  is split into two parts:

$$W_s = W_{iso} + W_{vol}$$

here,  $W_{iso}$  is the isochoric strain energy density and  $W_{vol}$  is the volumetric strain energy density.

The isochoric strain energy density can be entered as 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  $\overline{\epsilon}_{\text{el}}$ .

The volumetric strain energy density,  $W_{\text{vol}}$  can only be defined as an expression of the elastic volumetric deformation  $J_{\text{el}}$ . The quadratic volumetric strain energy density is defined as:

$$W_{\text{vol}}(J_{\text{el}}) = \frac{1}{2}\kappa(J_{\text{el}} - 1)^2 \quad (3-32)$$

and Hartmann–Neff (Ref. 10) volumetric strain energy as:

$$W_{\text{vol}}(J_{\text{el}}) = \frac{1}{50}\kappa(J_{\text{el}}^5 + J_{\text{el}}^{-5} - 2) \quad (3-33)$$

where  $\kappa$  is the bulk modulus. From here, the volumetric stress (pressure) is calculated as

$$p_{\text{m}} = -\frac{\partial W_{\text{vol}}}{\partial J}$$

When the expression in Equation 3-32 is used, the pressure becomes linearly related to the volume change:

$$p_{\text{m}} = -\kappa(J - 1)$$

but if the expression in Equation 3-33 is used instead of the quadratic function, the pressure becomes a nonlinear function of the volume change:

$$p_{\text{m}} = -\frac{1}{10}\kappa(J^4 - J^{-6})$$

Then, an auxiliary variable,  $p_{\text{w}}$ , is added to map the pressure when computing stresses.

The second Piola–Kirchhoff stress is then given by

$$S = -p_{\text{w}}JC^{-1} + 2\frac{\partial W_{\text{iso}}}{\partial C} \quad (3-34)$$

and the Cauchy stress tensor by

$$\sigma = J^{-1}FSF^T = -p_w I + 2J^{-1}F \frac{\partial W_{\text{iso}}}{\partial C} F^T$$

### INCOMPRESSIBLE HYPERELASTIC MATERIALS

When the **Incompressible material** is selected in the **Compressibility** list in the [Hyperelastic Material](#) node, the volumetric strain energy density  $W_{\text{vol}}$  is not defined as for [Nearly Incompressible Hyperelastic Materials](#), but instead, a weak constrain is added to account for the incompressibility condition

$$J_{\text{el}} = 1$$

The auxiliary variable,  $p_w$ , acts as Lagrange multiplier to enforce the constrain  $J_{\text{el}} = 1$ , this variable is then used as the pressure when computing stresses.

The strain energy density  $W_s$  consists only of the isochoric strain energy density  $W_{\text{iso}}$  contribution

$$W_s = W_{\text{iso}}$$

The second Piola–Kirchhoff stress is then given by

$$S = -p_w J C^{-1} + 2 \frac{\partial W_{\text{iso}}}{\partial C} \quad (3-35)$$

and the Cauchy stress tensor by

$$\sigma = J^{-1}FSF^T = -p_w I + 2J^{-1}F \frac{\partial W_{\text{iso}}}{\partial C} F^T \quad (3-36)$$

### 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. 12](#)). Finite elements in solid mechanics are said to “lock” when exhibiting an unphysical response to deformation ([Ref. 13](#)). 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 3-34](#) 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 done in the Linear Elastic Materials node.

### THEORY FOR THE PREDEFINED HYPERELASTIC MATERIAL MODELS

Different hyperelastic material models are constructed by specifying different elastic strain energy expressions. This module has several predefined material models and also has the option to enter user defined expressions for the strain energy density.

#### NEO-HOOKEAN

The strain energy density for the compressible 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).

$$W_s = \frac{1}{2}\mu(I_1 - 3) - \mu \ln(J_{el}) + \frac{1}{2}\lambda[\ln(J_{el})]^2$$

Here,  $\lambda$  and  $\mu$  are the Lamé parameters (SI unit: Pa).

The nearly incompressible version uses the isochoric invariant  $I_1(\bar{C}_{el})$  to define the isochoric strain energy density  $W_{iso}$ ,

$$W_{iso} = \frac{1}{2}\mu(\bar{I}_1 - 3)$$

and the elastic volume ratio  $J_{el}$  and the bulk modulus  $\kappa$  to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{el} = 1$ , see [Incompressible Hyperelastic Materials](#).



See also the description of the [Neo-Hookean](#) material model in the Solid Mechanics interface documentation.

---

#### ST VENANT-KIRCHHOFF

One of the simplest hyperelastic material models is the St Venant–Kirchhoff material, which is an extension of a linear elastic material into the hyperelastic regime.

The elastic strain energy density is written with two parameters (the two Lamé coefficients) and two invariants of the elastic Green–Lagrange strain tensor,  $I_1(\epsilon_{el})$  and  $I_2(\epsilon_{el})$

$$W_s = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2$$

Here,  $\lambda$  and  $\mu$  are Lamé parameters (SI unit: Pa). The bulk modulus  $\kappa$  is calculated from the Lamé parameters  $\kappa = \lambda + 2\mu/3$ .

The nearly incompressible version uses the isochoric invariants  $I_1(\overline{\epsilon}_{el})$  and  $I_2(\overline{\epsilon}_{el})$  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  $J_{e1}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#).



See also the description of the [St Venant–Kirchhoff](#) material model in the Solid Mechanics interface documentation.

#### MOONEY–RIVLIN, TWO PARAMETERS

For the nearly incompressible version, the isochoric strain energy density is written in terms of the two isochoric invariants of the elastic right Cauchy–Green deformation tensors  $I_1(\overline{C}_{el})$  and  $I_2(\overline{C}_{el})$

$$W_{iso} = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3)$$

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  $J_{e1}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, 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, 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

$$W_s = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} C_{mn} (I_1 - 3)^m (I_2 - 3)^n$$

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

$$W_s = C_{10}(I_1 - 3) + C_{01}(I_2 - 3)$$

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 nearly incompressible version uses the isochoric invariants of the elastic right Cauchy–Green deformation tensors  $I_1(\bar{C}_{el})$  and  $I_2(\bar{C}_{el})$  to define the isochoric strain energy density

$$W_{iso} = \sum_{m=0}^2 \sum_{n=0}^2 C_{mn} (\bar{I}_1 - 3)^m (\bar{I}_2 - 3)^n$$

The elastic volume ratio  $J_{el}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, 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



$$W_{\text{iso}} = \sum_{m=0}^3 \sum_{n=0}^3 C_{mn} (\bar{I}_1 - 3)^m (\bar{I}_2 - 3)^n$$

where  $I_1(\bar{C}_{\text{el}})$  and  $I_2(\bar{C}_{\text{el}})$  are the isochoric invariants of the elastic right Cauchy–Green deformation tensors. The elastic volume ratio  $J_{\text{el}}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density, see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{\text{el}} = 1$ , see [Incompressible Hyperelastic Materials](#).



See also the description of the [Mooney–Rivlin, Nine Parameters](#) material model in the Solid Mechanics interface documentation.

#### YE OH

Yeoh proposed ([Ref. 1](#)) a phenomenological model in order to fit experimental data of filled rubbers, where Mooney–Rivlin and Neo-Hookean models were too 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  $I_1(C_{\text{el}})$

$$W_{\text{s}} = c_1(I_1 - 3) + c_2(I_1 - 3)^2 + c_3(I_1 - 3)^3$$

The shear modulus depends on the deformation, and it is calculated as

$$\mu = 2 \left( \frac{\partial W_{\text{s}}}{\partial I_1} + \frac{\partial W_{\text{s}}}{\partial I_2} \right) = 2c_1 + 4c_2(I_1 - 3) + 6c_3(I_1 - 3)^2$$

This imposes a restriction on the coefficients  $c_1$ ,  $c_2$ ,  $c_3$ , since  $\mu > 0$ .

The nearly incompressible version uses the isochoric invariant of the elastic right Cauchy–Green deformation tensor  $I_1(\bar{C}_{\text{el}})$  to define the isochoric strain energy density

$$W_{\text{iso}} = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_1 - 3)^2 + c_3(\bar{I}_1 - 3)^3$$

The elastic volume ratio  $J_{\text{el}}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{\text{vol}}$ , see [Nearly Incompressible Hyperelastic Materials](#). The

incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{e1} = 1$ , 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

$$W_s = \sum_{p=1}^N \frac{\mu_p}{\alpha_p} (\lambda_{e1}^{\alpha_p} + \lambda_{e2}^{\alpha_p} + \lambda_{e3}^{\alpha_p} - 3)$$

Here  $\mu_p$  (SI unit: Pa) and  $\alpha_p$  (dimensionless) are material parameters, and  $\lambda_{e1}$ ,  $\lambda_{e2}$ , and  $\lambda_{e3}$  are the principal elastic stretches such as  $J_{e1} = \lambda_{e1}\lambda_{e2}\lambda_{e3}$ .

The Ogden model is empirical, in the sense that it does not relate the material parameters  $\alpha_p$  and  $\mu_p$  to physical phenomena. The parameters  $\alpha_p$  and  $\mu_p$  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 nearly incompressible version uses the isochoric elastic stretches

$$\bar{\lambda}_{eli} = \lambda_{eli} / J_{e1}^{1/3}$$

so the isochoric strain energy density is defined as

$$W_{iso} = \sum_{p=1}^N \frac{\mu_p}{\alpha_p} (\bar{\lambda}_{e1}^{\alpha_p} + \bar{\lambda}_{e2}^{\alpha_p} + \bar{\lambda}_{e3}^{\alpha_p} - 3)$$

The isochoric elastic stretches define a volume preserving deformation, since

$$\bar{\lambda}_{e1}\bar{\lambda}_{e2}\bar{\lambda}_{e3} = \lambda_{e1}\lambda_{e2}\lambda_{e3} / J_{e1} = 1$$

The shear modulus is then defined from

$$\mu = \frac{1}{2} \sum_{p=1}^{N} \alpha_p \mu_p$$

The elastic volume ratio  $J_{e1}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{e1} = 1$ , see [Incompressible Hyperelastic Materials](#).



Sometimes a slightly different definition of the strain energy function for the Ogden material is used. If you have material data for an Ogden material given, be careful that the definitions used are the same. If not, you will need to rescale the  $\mu_p$  coefficients.



See also the description of the [Ogden](#) material model in the Solid Mechanics interface documentation.

### STORAKERS

The Storakers material ([Ref. 13](#) and [Ref. 16](#)) is used to model highly compressible foams. The strain energy density is written in a similar fashion as in Ogden material:

$$W_s = \sum_{k=1}^N \frac{2\mu_k}{\alpha_k} \left( \lambda_{el1}^{\alpha_k} + \lambda_{el2}^{\alpha_k} + \lambda_{el3}^{\alpha_k} - 3 + \frac{1}{\beta_k} (J_{e1}^{-\alpha_k} \beta_k - 1) \right)$$

The initial shear and bulk moduli are computed from the parameters  $\mu_k$  (SI unit: Pa) and  $\beta_k$  (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

$$W_s = c_1(\lambda_{e11} + \lambda_{e12} + \lambda_{e13} - 3) + c_2(\lambda_{e11}\lambda_{e12} + \lambda_{e12}\lambda_{e13} + \lambda_{e11}\lambda_{e13} - 3)$$

The nearly incompressible version uses the isochoric elastic stretches defined as

$$\bar{\lambda}_{eli} = \lambda_{eli}/J_{el}^{1/3}$$

so the isochoric strain energy density is defined as

$$W_{iso} = c_1(\bar{\lambda}_{e11} + \bar{\lambda}_{e12} + \bar{\lambda}_{e13} - 3) + c_2(\bar{\lambda}_{e11}\bar{\lambda}_{e12} + \bar{\lambda}_{e12}\bar{\lambda}_{e13} + \bar{\lambda}_{e11}\bar{\lambda}_{e13} - 3)$$

The simplest Varga model is obtained by setting  $c_1 = \mu$  and  $c_2 = 0$ :

$$W_{iso} = \mu(\bar{\lambda}_{e11} + \bar{\lambda}_{e12} + \bar{\lambda}_{e13} - 3)$$

The elastic volume ratio  $J_{el}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, 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

$$W_s = \mu_0 \sum_{p=1}^{\infty} c_p (I_1^p - 3^p)$$

Here,  $\mu_0$  is the initial macroscopic shear modulus,  $I_1(C_{el})$  is the first invariant of the elastic right Cauchy–Green deformation tensor, and the coefficients  $c_p$  are obtained by series expansion of the inverse Langevin function.

Arruda and Boyce truncated the series and used only the first five terms of the series. The 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 N	11/1050 N <sup>2</sup>	19/7000 N <sup>3</sup>	519/673750 N <sup>4</sup>

Other authors (Ref. 1) use only the first three coefficients of the series. The number of segments in the polymeric chain is specified by the parameter  $N$  so the material model is described by only two parameters,  $\mu_0$  (SI unit: Pa) and  $N$ . This material model is sometimes also called the *eight-chain model*.

The nearly incompressible version uses the isochoric invariant  $\bar{I}_1 = I_1(\bar{C}_{el})$  to define the isochoric strain energy density

$$W_{iso} = \mu_0 \sum_{p=1}^5 c_p (\bar{I}_1^p - 3^p)$$

The elastic volume ratio  $J_{el}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, 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. Gent material (Ref. 14 and Ref. 15) is a simple phenomenological constitutive model based on only two parameters,  $\mu$  and  $j_m$ , which defines the strain energy density as:

$$W_s = -\frac{\mu}{2} j_m \log\left(1 - \frac{\bar{I}_1 - 3}{j_m}\right)$$

Here,  $\mu$  (SI unit: Pa) is the shear modulus and  $j_m$  (dimensionless) is a limiting value for  $\bar{I}_1 - 3$ , which takes care of the limiting polymeric chain extensibility of the material.

Since the strain energy density does not depend on the second invariant  $I_2$ , Gent model is often classified as a generalized Neo-Hookean material. The strain energy density tends to be the one of incompressible Neo-Hookean material as  $j_m \rightarrow \infty$ .

The nearly incompressible formulation uses the isochoric invariants  $I_1(\bar{C}_{el})$  to define the isochoric strain energy density

$$W_{iso} = -\frac{\mu}{2} j_m \log\left(1 - \frac{\bar{I}_1 - 3}{j_m}\right)$$

The elastic volume ratio  $J_{el}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{vol}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{el} = 1$ , see [Incompressible Hyperelastic Materials](#). Gent material is the simplest model of the limiting chain extensibility family.



See also the description of the [Gent](#) 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 valid for compressible isotropic hyperelastic materials (Ref. 1).

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  $I_3(C_{el})$

$$W_s = \phi \frac{\mu}{2} (I_1 - 3) + \frac{1}{\beta} (I_3^\beta - 1) + (1 - \phi) \frac{\mu}{2} \left( \left( \frac{I_2}{I_3} - 3 \right) + \frac{1}{\beta} (I_3^\beta - 1) \right)$$

Here,  $\phi$  is an interpolation parameter bounded to  $0 < \phi < 1$ , the parameter  $\mu$  (SI unit: Pa) is the shear modulus, and  $\beta$  is an expression of Poisson's ratio.

When the parameter  $\beta \rightarrow \infty$ , the strain energy simplifies to a similar form of the Mooney–Rivlin material 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 similar form of the Neo-Hookean model

$$W_s = \frac{\mu}{2} (I_1 - 3) + \frac{\mu}{2\beta} (J^{-2\beta} - 1)$$



See also the description of the [Blatz–Ko](#) material model in the Solid Mechanics interface documentation.

---

#### GAO

Gao proposed (Ref. 17) a simple 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:

$$I_{-1} = \text{trace}(C_{el}^{-1}) = \frac{I_2(C_{el})}{I_3(C_{el})}$$

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:

$$E = \frac{3^n n^2 8a}{2n + 1} \quad \text{and} \quad \nu = \frac{n - 1}{2n + 1}$$

Since  $n = (1+\nu)/(1-2\nu)$  and it is bounded to  $1 < n < 3$ , this material model is stable for materials with 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 potential is used in nonlinear acoustoelasticity. Most conveniently, it is expressed in terms of the three invariants of the elastic Green–Lagrange strain tensor,  $I_1(\epsilon_{el})$ ,  $I_2(\epsilon_{el})$ , and  $I_3(\epsilon_{el})$ :

$$W_s = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2 + \frac{1}{3}(l + 2m)I_1^3 - 2mI_1I_2 + nI_3$$

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  $\lambda$  and  $\mu$  are the Lamé parameters.



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. 19](#)). The strain energy density is defined by two parameters,  $a$  (SI unit: Pa) and  $b$ , and the first invariant of the isochoric elastic right Cauchy–Green deformation tensors  $I_1(\bar{C}_{el})$ :

$$W_{iso} = \frac{a}{b} \left( e^{\frac{b}{2}(\bar{I}_1 - 3)} - 1 \right)$$

The parameter  $a$  plays the role of shear modulus at the small strain limit.

The incompressible option uses the isochoric strain energy, and an extra variable is added to enforce the incompressibility condition  $J_{el} = 1$ , see [Incompressible Hyperelastic Materials](#).



## FUNG

Fung (Ref. 19) proposed one of the most popular strain energy functions to describe the deformation in soft biological tissues and arteries. For this hyperelastic material, the strain energy density reads

$$W_s = \frac{c}{2}(e^Q - 1)$$

the parameter  $c$  (SI unit: Pa) scales the global stiffness, and the quadratic form  $Q$  (dimensionless) depends on the Green–Lagrange strain

$$Q = \mathbf{E}^T \cdot \mathbf{A} \cdot \mathbf{E}$$

here,  $\mathbf{A}$  is a six by six 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  $\mathbf{A}$  is described by nine independent parameters (Ref. 19), see [Orthotropic Material](#) to see the structure of such matrix.

The nearly incompressible version uses the components of the isochoric elastic Green–Lagrange strain  $\bar{\mathbf{E}}_{e1}$  to define the isochoric strain energy density

$$W_{\text{iso}} = \frac{c}{2}(e^{\bar{Q}} - 1)$$

where

$$\bar{Q} = \bar{\mathbf{E}}^T \cdot \mathbf{A} \cdot \bar{\mathbf{E}}$$

The elastic volume ratio  $J_{e1}$  and the bulk modulus  $\kappa$  are used to define the volumetric strain energy density  $W_{\text{vol}}$ , see [Nearly Incompressible Hyperelastic Materials](#). The incompressible option uses the same isochoric strain energy as when selecting the nearly incompressible option, but an extra variable is added to enforce the incompressibility condition  $J_{e1} = 1$ , see [Incompressible Hyperelastic Materials](#).

## EXTENDED TUBE

The extended tube model is a micro-mechanics inspired material model that considers the network constraints from molecular chains and the limited chain extensibility (Ref. 20-22). The isochoric strain energy density of the model consists of two terms

$$W_{\text{iso}} = W_c + W_e$$

The term  $W_c$  represents the energy from the cross-linking of the network, and  $W_e$  represents the energy from confining tube constrains. These are written as

$$W_c = \frac{G_c}{2} \left( \frac{(1-\alpha)(\bar{I}_1 - 3)}{1-\alpha(\bar{I}_1 - 3)} + \log(1-\alpha(\bar{I}_1 - 3)) \right)$$

and

$$W_e = \frac{2G_e}{\beta^2} (\bar{\lambda}_{\text{el}1}^{-\beta} + \bar{\lambda}_{\text{el}2}^{-\beta} + \bar{\lambda}_{\text{el}3}^{-\beta} - 3)$$

Here,  $G_c$  and  $G_e$  are material parameters (SI unit: Pa),  $\bar{I}_1 = I_1(\bar{C}_{\text{el}})$  is the first invariant of the elastic right Cauchy–Green deformation tensor,  $\bar{\lambda}_{\text{el}}$  are the isochoric principal stretches, and  $\alpha$  and  $\beta$  are dimensionless coefficients.

The parameters  $G_c$  and  $G_e$  plays the role of shear modulus at the small strain limit,  $G_0 = G_c + G_e$ . The parameter  $\alpha$  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 incompressible option uses the isochoric strain energy, and it adds an extra variable to enforce the incompressibility condition  $J_{\text{el}} = 1$ , see [Incompressible Hyperelastic Materials](#). Note that the material might be unstable for certain combinations of material parameters, see [Ref. 20-21](#) 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  $C_{\text{el}}$ , the elastic right Cauchy–Green deformation tensor in the local material coordinate system.
- Principal invariants of  $C_{\text{el}}$

$$I_1(C_{\text{el}}) = \text{trace}(C_{\text{el}})$$

$$I_2(C_{\text{el}}) = \frac{1}{2}(I_1^2(C_{\text{el}}) - \text{trace}(C_{\text{el}}^2))$$

$$I_3(C_{el}) = \det(C_{el})$$



The internal variables for these invariants are named `solid.I1Cel`, `solid.I2Cel`, and `solid.I3Cel`.

- Components of the elastic Green–Lagrange strain tensor  $\epsilon_{el}$  in the local coordinate system.
- Principal elastic stretches  $\lambda_{el1}$ ,  $\lambda_{el2}$ , and  $\lambda_{el3}$ , 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

$$\epsilon_{el} = \frac{1}{2}(C_{el} - \mathbf{I})$$

the invariants of  $\epsilon_{el}$  are written in terms of the invariants of  $C_{el}$ :

$$I_1(\epsilon_{el}) = \text{trace}(\epsilon_{el}) = \frac{1}{2}(I_1(C_{el}) - 3)$$

$$I_2(\epsilon_{el}) = \frac{1}{2}(I_1^2(\epsilon_{el}) - \text{trace}(\epsilon_{el}^2)) = \frac{1}{4}(I_2(C_{el}) - 2I_1(C_{el}) + 3)$$

$$I_3(\epsilon_{el}) = \det(\epsilon_{el}) = \frac{1}{8}(I_3(C_{el}) - I_2(C_{el}) + I_1(C_{el}) - 1)$$



The internal variables for these invariants are named `solid.I1eel`, `solid.I2eel`, and `solid.I3eel`.

When the **Nearly incompressible material** option is selected for the [Hyperelastic Material](#) node, 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  $J_{el} = \det(F_{el})$

- The isochoric strain energy,  $W_{\text{iso}}$ , as an expression involving the invariants of the isochoric elastic right Cauchy–Green tensor  $I_1(\overline{C}_{\text{el}})$  and  $I_2(\overline{C}_{\text{el}})$ ; the invariants of the isochoric elastic Green–Lagrange strain  $I_1(\overline{\epsilon}_{\text{el}})$ ,  $I_2(\overline{\epsilon}_{\text{el}})$ , and  $I_3(\overline{\epsilon}_{\text{el}})$ ; or the principal isochoric elastic stretches defined as  $\bar{\lambda}_{\text{cli}} = \lambda_{\text{cli}}/J_{\text{el}}^{1/3}$



The internal variables for  $J_{\text{el}}$ ,  $I_1(\overline{C}_{\text{el}})$ , and  $I_2(\overline{C}_{\text{el}})$  are named `solid.Je1`, `solid.I1C1e1`, and `solid.I2C1e1`.

The internal variables for  $I_1(\overline{\epsilon}_{\text{el}})$ ,  $I_2(\overline{\epsilon}_{\text{el}})$ , and  $I_3(\overline{\epsilon}_{\text{el}})$  are named `solid.I1e1e1`, `solid.I2e1e1`, and `solid.I3e1e1`.

When the **Incompressible material** option is selected, enter an expression for the isochoric elastic strain energy  $W_{\text{iso}}$ , as done for the **Nearly incompressible material** option. An extra variable is added to enforce the incompressibility condition  $J_{\text{el}} = 1$ , see [Incompressible Hyperelastic Materials](#).



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  $\mathbf{F} = \nabla \mathbf{u} + \mathbf{I}$  tensors, their transpose, inversions, as well as the global material system components of  $\mathbf{C}$  and  $\epsilon$ . If they occur, such variables are treated as constants during symbolic differentiations.



See also the description of the [User Defined](#) material model in the Solid Mechanics interface documentation.

### 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. 27-29](#)) describes the stress-softening phenomenon observed under cyclic loading in elastomers and biological materials.

Ogden and Roxburgh ([Ref. 30](#)) used an additional state variable to model the Mullins effect. The state variable  $\eta$  is introduced to memorize the microstructural damage on reinforced rubber after repeated loading-unloading cycles. The modified isochoric strain energy density reads

$$\bar{W}_{\text{iso}} = \eta W_{\text{iso}} + \phi(\eta)$$

here,  $W_{\text{iso}}$  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 the some constrains are fulfilled. The authors proposed a state function based on the error function, which defined how the state variable  $\eta$  varies as a function of the isochoric strain energy

$$\eta = 1 - \frac{1}{r} \operatorname{erf}\left(\frac{W_{\text{max}} - W_{\text{iso}}}{m}\right)$$

here,  $\operatorname{erf}(\cdot)$  is the error function,  $r$  and  $m$  are positive parameters, and  $W_{\text{max}}$  is the maximum attained value of the isochoric strain energy density on the loading path.

The associated microstructural damage is computed from

$$d = 1 - \eta = \frac{1}{r} \operatorname{erf}\left(\frac{W_{\text{max}} - W_{\text{iso}}}{m}\right)$$

and the isochoric strain energy modified by the Mullins effect reads

$$\bar{W}_{\text{iso}} = (1 - d)W_{\text{iso}}$$

Over the years others authors (Ref. 31-32) have proposed different flavors of Ogden–Roxburgh model, the version implemented in COMSOL Multiphysics uses the hyperbolic tangent function instead of the error function, and a parameter to define the maximum allowed damage  $d_{\infty}=1/r$ . The microstructural damage is then computed from

$$d = d_{\infty} \tanh\left(\frac{W_{\text{max}} - W_{\text{iso}}}{W_{\text{sat}} + \alpha W_{\text{max}}}\right)$$

Here,  $d_{\infty}$ ,  $\alpha$ , and  $W_{\text{sat}}$  are positive parameters.

Miehe (Ref. 33) proposed an exponential expression for the damage variable in order to model the Mullins effect

$$d = d_{\infty}(1 - e^{-W_{\text{max}}/W_{\text{sat}}})$$

where  $W_{\text{sat}}$  and  $d_{\infty}$  are positive parameters.

In this section:

- [Introduction](#)
- [Defining the Yield Surface](#)
- [Isotropic Plasticity](#)
- [Yield Function](#)
- [Hill Orthotropic Plasticity](#)
- [Soil Plasticity](#)
- [Porous Plasticity](#)
- [Isotropic Hardening](#)
- [Kinematic Hardening](#)
- [Introduction to Small and Large Plastic Strains](#)
- [Plastic Flow for Small Strains](#)
- [Plastic Flow for Large Strains](#)
- [Numerical Solution of the Elastoplastic Conditions](#)
- [Energy Dissipation](#)

### **INTRODUCTION**

Many materials have a distinct elastic regime, in which the deformations are recoverable and path independent. When the stresses exceed a certain level, the *yield limit*, permanent plastic strains will appear.

The elastic part of the constitutive relation can be described by either a [Linear Elastic Material](#), [Nonlinear Elastic Materials](#), or by [Hyperelastic Material Models](#).

Elastoplastic material models are common, both when modeling metals and soils.

In geotechnical applications it is common to define compressive stresses as having positive signs. In COMSOL Multiphysics, the convention is however to always use positive signs for tensile stresses.



See also [Sign Conventions](#).

---

## DEFINING THE YIELD SURFACE

A yield criterion serves to define the stress condition under which plastic deformation occurs. Stress paths inside the yield surface result in purely recoverable deformations (elastic behavior), while paths intersecting the yield surface produces both recoverable and permanent deformations (plastic strains).

In general, the yield surface can be described as

$$F = f(\sigma) - f_c = 0$$

where  $f_c$  can be a constant value (for perfectly plastic materials), or a variable for strain-hardening materials. The yield surface  $F$  is a surface in the space of principal stresses, in which the elastic regime ( $F \leq 0$ ) is enclosed.

For brittle materials, the yield surface represents a *failure surface*, which is a stress level at which the material collapses instead of deforms plastically.



Some authors define the *yield criterion* as  $f(\sigma) = f_c$ , while the yield surface is an isosurface in the space of principal stresses  $F = 0$ , which can be chosen for numerical purposes as  $F = f(\sigma)^2 - f_c^2 = 0$ .

## ISOTROPIC PLASTICITY

For isotropic plasticity, the plastic potential  $Q_p$  is written in terms of at most three invariants of Cauchy's stress tensor

$$Q_p(\sigma) = Q_p(I_1(\sigma), J_2(\sigma), J_3(\sigma))$$

where the invariants of the stress tensor are

$$I_1(\sigma) = \text{trace}(\sigma)$$

$$J_2(\sigma) = \frac{1}{2} \text{dev}(\sigma) : \text{dev}(\sigma)$$

$$J_3(\sigma) = \det(\text{dev}(\sigma))$$

so that the increment of the plastic strain tensor  $\dot{\epsilon}_p$  can be decomposed into

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma} = \lambda \left( \frac{\partial Q_p}{\partial I_1} \frac{\partial I_1}{\partial \sigma} + \frac{\partial Q_p}{\partial J_2} \frac{\partial J_2}{\partial \sigma} + \frac{\partial Q_p}{\partial J_3} \frac{\partial J_3}{\partial \sigma} \right)$$

The increment in the plastic strain tensor  $\dot{\epsilon}_p$  includes in a general case both deviatoric and volumetric parts. The tensor  $\dot{\epsilon}_p$  is symmetric given the following properties

$$\begin{aligned}
\frac{\partial I_1}{\partial \sigma} &= I \\
\frac{\partial J_2}{\partial \sigma} &= \text{dev}(\sigma) \\
\frac{\partial J_3}{\partial \sigma} &= \text{dev}(\sigma)\text{dev}(\sigma) - \frac{2}{3}J_2I
\end{aligned}
\tag{3-37}$$

The trace of the incremental plastic strain tensor, which is called the *volumetric plastic strain rate*  $\dot{\epsilon}_{\text{pvol}}$ , is only a result of dependence of the plastic potential on the first invariant  $I_1(\sigma)$ , since  $\partial J_2/\partial \sigma$  and  $\partial J_3/\partial \sigma$  are deviatoric tensors

$$\dot{\epsilon}_{\text{pvol}} = \text{trace}(\dot{\epsilon}_p) = \lambda \text{trace}\left(\frac{\partial Q_p}{\partial \sigma}\right) = 3\lambda \frac{\partial Q_p}{\partial I_1}$$

A common measure of inelastic deformation is the *equivalent plastic strain rate*, which is defined from the plastic shear components

$$\dot{\epsilon}_{\text{pe}} = \sqrt{\frac{2}{3}\text{dev}(\dot{\epsilon}_p):\text{dev}(\dot{\epsilon}_p)} \tag{3-38}$$

For metal plasticity under the von Mises or Tresca criteria, the volumetric plastic strain rate  $\dot{\epsilon}_{\text{pvol}}$  is always zero because the plastic potential is independent of the invariant  $I_1(\sigma)$ . This is known as *J2 plasticity*.

Incompressible plastic deformation is experimentally observed in metals, but this is not the case for most materials used in geotechnical applications. For instance, a nonzero volumetric plastic strain is explicitly used in [Porous Plasticity](#) and [Elastoplastic Soil Models](#).



- The equivalent plastic strain and the volumetric plastic strain are available in the variables `solid.epe` and `solid.epvol`.
- In a time-dependent analysis, rates of plastic strains can be computed with expressions like `d(solid.epe, TIME)`.

### YIELD FUNCTION

When an associated flow rule is applied, the yield function must be smooth, that is, continuously differentiable with respect to the stress. In COMSOL Multiphysics, the following form is used:



$$F_y = \phi(\sigma) - \sigma_{ys}$$

where  $\sigma_{ys}$  is the yield stress. The scalar function  $\phi(\sigma)$  is called equivalent stress. The default form of the equivalent stress is the *von Mises stress*, which is often used in metal plasticity:

$$\sigma_{mises} = \sqrt{3J_2(\sigma)} = \sqrt{\frac{3}{2}\text{dev}(\sigma):\text{dev}(\sigma)}$$

Other expressions can be defined, such as *Tresca stress*, *Hill orthotropic plasticity*, or a user defined expression.

The Tresca equivalent stress is calculated from the difference between the largest and the smallest principal stress

$$\sigma_{tresca} = \sigma_{p1} - \sigma_{p3}$$

A user defined yield function can be expressed in terms of invariants of the stress tensor such as the pressure (volumetric stress)

$$p = -\frac{1}{3}I_1(\sigma)$$

the equivalent (von Mises) stress  $\sigma_{mises}$ , or other invariants, principal stresses, or stress tensor components.

#### The von Mises Criterion

The von Mises criterion suggests that the yielding of the material begins when the second deviatoric stress invariant  $J_2$  reaches a critical value. This criterion can be written in terms of the elements of Cauchy's stress tensor ([Ref. 1](#))

$$J_2 = \frac{1}{6}((\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2) + \sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2 = k^2$$

The von Mises criterion is implemented with the yield surface

$$F = \sqrt{3J_2} - \sigma_{ys} = 0$$

where  $\sigma_{ys}$  is the *yield stress level* (yield stress in uniaxial tension).



The equivalent or von Mises stress ( $\sigma_{mises} = \sqrt{3J_2}$ ) is available in the variable `solid.mises`, where `solid` is the name of the physics interface node.

---

### The Tresca Criterion

The Tresca yield surface is normally expressed in terms of the principal stress components

$$\frac{1}{2} \max(|\sigma_1 - \sigma_2|, |\sigma_1 - \sigma_3|, |\sigma_2 - \sigma_3|) = k$$

The Tresca criterion is a hexagonal prism with its axis equally inclined to the three principal stress axes. When the principal stresses fulfill  $\sigma_1 \geq \sigma_2 \geq \sigma_3$ , this criterion is written as

$$\frac{1}{2}(\sigma_1 - \sigma_3) = k$$

By using the representation of principal stresses in term of the invariants  $J_2$  and the Lode angle  $0 \leq \theta \leq \pi/3$ , this criterion can alternatively be written as

$$\frac{1}{2} \sqrt{\frac{4J_2}{3}} \left( \cos(\theta) - \cos\left(\theta + \frac{2\pi}{3}\right) \right) = \sqrt{J_2} \sin\left(\theta + \frac{\pi}{3}\right) = k$$

or equivalently

$$\sqrt{J_2} \cos\left(\theta - \frac{\pi}{6}\right) = k$$

The maximum shear stress is reached at the meridians ( $\theta = 0$  or  $\theta = \pi/3$ ). The Tresca criterion can be circumscribed by setting the Lode angle  $\theta = 0$ , or equivalently, by a von Mises criterion

$$\sqrt{3J_2} = 2k$$

The minimum shear is reached at  $\theta = \pi/6$ , so the Tresca criterion can be inscribed by setting a von Mises criterion

$$\sqrt{J_2} = k$$

When dealing with soils, the parameter  $k$  is also called *undrained shear strength*.

The Tresca criterion can be used with either an associated or nonassociated flow rule, in which case von Mises stress is applied in order to get better numerical performance.



The Tresca equivalent stress,  $\sigma_{\text{tresca}} = \sigma_1 - \sigma_3$ , is implemented in the variable `solid.tresca`, where `solid` is the name of the physics interface node.

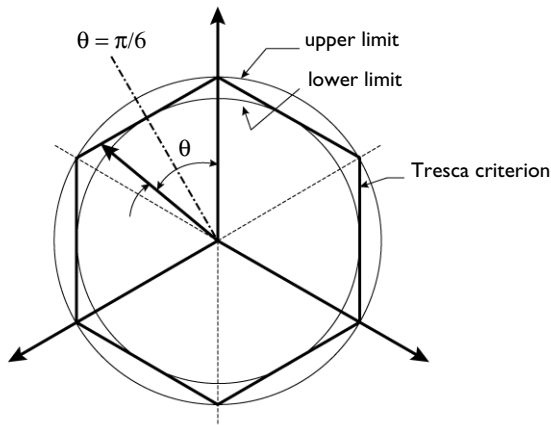


Figure 3-14: The upper and lower limits of the Tresca criterion.

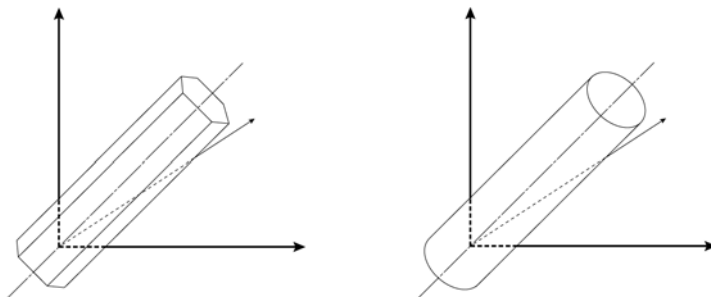


Figure 3-15: Classical yield criteria for metals. The Tresca criterion (left) and the von Mises criterion (right).

The von Mises and Tresca criteria are independent of the first stress invariant  $I_1$  and are mainly used for the analysis of plastic deformation in metals and ductile materials, though some researchers also use these criteria for describing fully saturated cohesive soils under undrained conditions. The von Mises and Tresca criteria belong to what researchers call *volume preserving* or  $J_2$  plasticity, as the plastic flow is independent on the mean pressure.

#### HILL ORTHOTROPIC PLASTICITY

Hill (Ref. 6, Ref. 7) proposed a quadratic yield function (and associated plastic potential) in a local coordinate system given by the principal axes of orthotropy  $\mathbf{a}_i$

$$\begin{aligned} Q_p = & F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + \\ & 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 - 1 \end{aligned} \quad (3-39)$$

The six parameters  $F$ ,  $G$ ,  $H$ ,  $L$ ,  $M$ , and  $N$  are related to the state of anisotropy. As with isotropic plasticity, the elastic region  $Q_p < 0$  is bounded by the yield surface  $Q_p = 0$ .

Hill demonstrated that this type of anisotropic plasticity is volume preserving, this is, given the associated flow rule

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma}$$

the trace of the plastic strain rate tensor is zero, which follows from the expressions for the diagonal elements of  $\dot{\epsilon}_p$

$$\begin{aligned} \dot{\epsilon}_{p11} &= \lambda \frac{\partial Q_p}{\partial \sigma_{11}} = 2\lambda(-G(\sigma_{33} - \sigma_{11}) + H(\sigma_{11} - \sigma_{22})) \\ \dot{\epsilon}_{p22} &= \lambda \frac{\partial Q_p}{\partial \sigma_{22}} = 2\lambda(F(\sigma_{22} - \sigma_{33}) - H(\sigma_{11} - \sigma_{22})) \\ \dot{\epsilon}_{p33} &= \lambda \frac{\partial Q_p}{\partial \sigma_{33}} = 2\lambda(-F(\sigma_{22} - \sigma_{33}) + G(\sigma_{33} - \sigma_{11})) \end{aligned}$$

so the plastic volumetric strain rate is zero

$$\dot{\epsilon}_{p\text{vol}} = \text{trace}(\dot{\epsilon}_p) = \dot{\epsilon}_{p11} + \dot{\epsilon}_{p22} + \dot{\epsilon}_{p33} = 0$$



Hill plasticity is an extension of  $J2$  (von Mises) plasticity, in the sense that it is volume preserving. Due to this assumption, six parameters are needed to define orthotropic plasticity, as opposed to orthotropic elasticity, where nine elastic coefficients are needed.

*Expressions for the Coefficients  $F$ ,  $G$ ,  $H$ ,  $L$ ,  $M$ ,  $N$*

Hill noticed that the parameters  $L$ ,  $M$ , and  $N$  are related to the yield stress in shear with respect to the axes of orthotropy  $\mathbf{a}_i$ , thus they are positive parameters

$$L = \frac{1}{2\sigma_{ys23}^2}, \quad M = \frac{1}{2\sigma_{ys31}^2}, \quad N = \frac{1}{2\sigma_{ys12}^2}$$

Here,  $\sigma_{ysij}$  represents the yield stress in shear on the plane  $ij$ .

The material parameters  $\sigma_{ys1}$ ,  $\sigma_{ys2}$ , and  $\sigma_{ys3}$  represent the tensile yield stress in the direction,  $\mathbf{a}_1$ ,  $\mathbf{a}_2$ , and  $\mathbf{a}_3$ , and they are related to Hill's parameters  $F$ ,  $G$ , and  $H$  as

$$\begin{aligned} \frac{1}{\sigma_{ys1}^2} &= G + H \\ \frac{1}{\sigma_{ys2}^2} &= H + F \\ \frac{1}{\sigma_{ys3}^2} &= F + G \end{aligned}$$

or equivalently

$$\begin{aligned} 2F &= \frac{1}{\sigma_{ys2}^2} + \frac{1}{\sigma_{ys3}^2} - \frac{1}{\sigma_{ys1}^2} \\ 2G &= \frac{1}{\sigma_{ys3}^2} + \frac{1}{\sigma_{ys1}^2} - \frac{1}{\sigma_{ys2}^2} \\ 2H &= \frac{1}{\sigma_{ys1}^2} + \frac{1}{\sigma_{ys2}^2} - \frac{1}{\sigma_{ys3}^2} \end{aligned}$$

Note that at most one of the three coefficients  $F$ ,  $G$ , and  $H$  can be negative.



In case of hardening, these coefficients (either Hill's coefficients or the shear and tensile yield stresses) are renamed with the "initial" prefix.

In order to define a yield function and plastic potential suitable for isotropic or kinematic hardening, the average initial yield stress  $\sigma_{ys0}$  is calculated from the Hill's parameters  $F$ ,  $G$ , and  $H$  (this is equivalent to the initial yield stress  $\sigma_{ys0}$  in von Mises plasticity)

$$\frac{1}{\sigma_{ys0}^2} = \frac{2}{3}(F + G + H) = \frac{1}{3} \left( \frac{1}{\sigma_{ys1}^2} + \frac{1}{\sigma_{ys2}^2} + \frac{1}{\sigma_{ys3}^2} \right) \quad (3-40)$$

Defining *Hill's equivalent stress* as (Ref. 7)

$$\begin{aligned} \sigma_{hill}^2 = & \sigma_{ys0}^2 (F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 \\ & + 2(L\sigma_{23}^2 + M\sigma_{31}^2 + N\sigma_{12}^2)) \end{aligned}$$

makes it possible to write the plastic potential in a similar way to von Mises plasticity.

Isotropic hardening is then applied on the average yield stress variable  $\sigma_{ys0}$ , by using the plastic potential

$$Q_p = \sigma_{hill} - \sigma_{ys}$$

Here, the average yield stress

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\epsilon_{pe})$$

now depends on the initial yield stress  $\sigma_{ys0}$ , the hardening function  $\sigma_h$ , and the equivalent plastic strain  $\epsilon_{pe}$ .

## SOIL PLASTICITY

### *The Mohr–Coulomb Criterion*

The Mohr–Coulomb criterion is the most popular criterion in soil mechanics. It was developed by Coulomb before the Tresca and von Mises criteria for metals, and it was the first criterion to account for 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

$$|\tau| + \sigma \tan \phi - c = 0$$

here,  $\tau$  is the shear stress,  $c$  the cohesion, and  $\phi$  denotes the angle of internal friction.

With the help of Mohr's circle, this 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 the space of principal stresses, which generates singularities in the derivatives of the yield function.

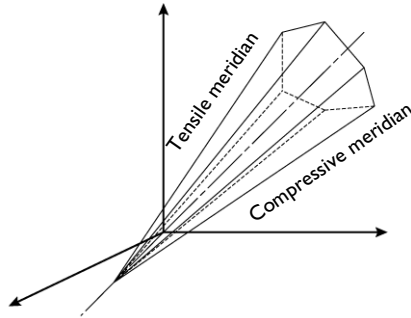


Figure 3-16: The Mohr–Coulomb criterion. The cone opens toward the compressive axis.

The Mohr–Coulomb criterion can be written in terms of the invariants  $I_1$  and  $J_2$  and the Lode angle  $0 \leq \theta \leq \pi/3$  (Ref. 1, Ref. 11) when the principal stresses are sorted as  $\sigma_1 \geq \sigma_2 \geq \sigma_3$ . The yield function then reads

$$F_y = \frac{1}{3}I_1 \sin \phi + \sqrt{\frac{J_2}{3}} \left( (1 + \sin \phi) \cos \theta - (1 - \sin \phi) \cos \left( \theta + \frac{2\pi}{3} \right) \right) - c \cos \phi = 0$$

The tensile meridian is defined when  $\theta = 0$  and the compressive meridian when  $\theta = \pi/3$ .

Rearranging terms, the Mohr–Coulomb criterion reads

$$F_y = \sqrt{J_2} m(\theta) + \alpha I_1 - k = 0$$

where

$$m(\theta) = \sqrt{\frac{1}{3}} \left( (1 + \sin \phi) \cos \theta - (1 - \sin \phi) \cos \left( \theta + \frac{2\pi}{3} \right) \right), \quad \alpha = \sin \phi / 3, \text{ and}$$

$$k = c \cos \phi$$

In the special case of frictionless material, ( $\phi = 0$ ,  $\alpha = 0$ ,  $k = c$ ), the Mohr–Coulomb criterion reduces to a Tresca’s maximum shear stress criterion,  $(\sigma_1 - \sigma_3) = 2k$  or equivalently

$$F_y = \sqrt{J_2} \cos \left( \theta - \frac{\pi}{6} \right) - k = 0$$



See also the description of the [Mohr–Coulomb](#) material model in the Solid Mechanics interface documentation.

---

#### *The Drucker–Prager Criterion*

The Mohr–Coulomb criterion causes numerical difficulties when treating the plastic flow at the corners of the yield surface. The Drucker–Prager model ([Ref. 1](#), [Ref. 2](#)) neglects the influence of the invariant  $J_3$  (introduced by the Lode angle) on the cross-sectional shape of the yield surface. It can be considered as the first attempt to approximate the Mohr–Coulomb criterion by a smooth function based on the invariants  $I_1$  and  $J_2$  together with two material constants (which can be related to Mohr–Coulomb’s coefficients)

$$F_y = \sqrt{J_2} + \alpha I_1 - k = 0$$



The criterion is sometimes also called the extended von Mises criterion, since it is equivalent to the von Mises criterion for metals when setting  $\alpha = 0$ .

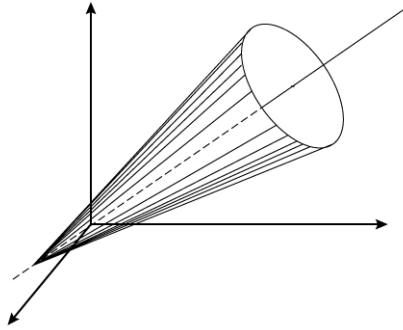


Figure 3-17: The Drucker–Prager criterion. The cone opens toward the compressive axis.

The coefficients in the Drucker–Prager model are related to the cohesion  $c$  and angle of internal friction  $\phi$  in Mohr–Coulomb criterion by the relation

$$\frac{\alpha}{k} = \frac{\tan \phi}{3c}$$

The coefficients in the Drucker–Prager model can be matched to the coefficients in the Mohr–Coulomb criterion by (Ref. 1)

$$\alpha = \frac{2}{\sqrt{3}} \cdot \frac{\sin \phi}{(3 \pm \sin \phi)} \quad \text{and} \quad k = \frac{2\sqrt{3}c \cos \phi}{(3 \pm \sin \phi)}$$

The symbol  $\pm$  is related to either matching the tensile meridian (positive sign) or the compressive meridian (negative sign) of Mohr–Coulomb’s pyramid.

The matching at the tensile meridian ( $\theta = 0$ ) comes from setting

$$m(0) = (3 + \sin \phi)/(2\sqrt{3})$$

in the Mohr–Coulomb criterion, and the matching at the compressive meridian ( $\theta = \pi/3$ ) from setting

$$m(\pi/3) = (3 - \sin \phi)/(2\sqrt{3})$$

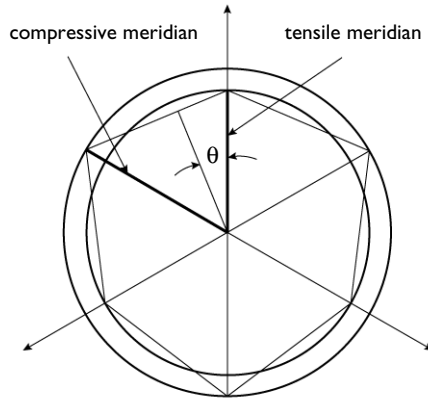


Figure 3-18: The Drucker–Prager criterion showing the tensile and compressive meridians (inner and outer circles), and the Lode angle compared to the cross section of the Mohr–Coulomb criterion in the  $\pi$ -plane.

In the special case of frictionless material, ( $\phi = 0$ ,  $\alpha = 0$ ,  $k = 2c/\sqrt{3}$ ), the Drucker–Prager criterion reduces to the von Mises criterion

$$\sqrt{J_2} = 2c/\sqrt{3}$$

When the Drucker–Prager criterion is matched to the Mohr–Coulomb criterion in 2D plane-strain applications, the parameters are

$$\alpha = \frac{\tan \phi}{\sqrt{9 + 12 \tan^2 \phi}} \quad \text{and} \quad k = \frac{3c}{\sqrt{9 + 12 \tan^2 \phi}}$$

when matching the criterion in 2D generalized plane-strain, the matching parameters are:

$$\alpha = \frac{2 \sin \phi}{\sqrt{3}(3 - \sin \phi)} \quad \text{and} \quad k = \frac{2\sqrt{3}}{3 - \sin \phi} \cdot c \cos \phi$$

and when matching the criterion in 2D plane-stress, the matching parameters are:

$$\alpha = \frac{1}{\sqrt{3}} \cdot \sin \phi \quad \text{and} \quad k = \frac{2}{\sqrt{3}} \cdot c \cos \phi$$

### *Dilatation Angle*

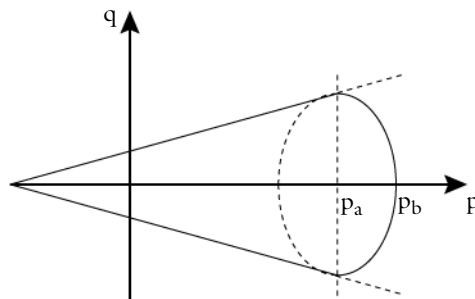
The Mohr–Coulomb yield criterion is sometimes used with a nonassociated plastic potential. This plastic potential could be either a Drucker–Prager criterion, or the same Mohr–Coulomb yield function but with a different slope with respect to the hydrostatic axis, in which case the angle of internal friction is replaced by the dilatation angle, which is normally smaller (Ref. 7).

Also, when using a Drucker–Prager criterion matched to a Mohr–Coulomb criterion, the plastic potential could also be non-associated, in which case the difference between the dilatation angle and the angle of internal friction would result in a yield surface and plastic potential portrayed by two cones with different angles with respect to the hydrostatic angle.

### *Elliptic Cap*

The Mohr–Coulomb and Drucker–Prager criteria portray a conic yield surface which opens in the hydrostatic axis direction. Normally, these soil models are not accurate above a given limit pressure because real-life materials cannot bear infinite loads and still behave elastically. A simple way to overcome this problem is to add an elliptical end cap on the compressive side to these models.

The elliptic cap is an elliptical yield surface of semi-axes as shown in Figure 3-19. The initial pressure  $p_a$  (SI units: Pa) denotes the pressure at which the elastic range circumscribed by either a Mohr–Coulomb pyramid or a Drucker–Prager cone is not valid any longer, so a cap surface is added. The limit pressure  $p_b$  gives the curvature of the ellipse, and denotes the maximum admissible hydrostatic pressure for which the material starts deforming plastically. Pressures higher than  $p_b$  are not allowed.



*Figure 3-19: Elliptic cap model in Haigh–Westergaard coordinate system.*

The sign convention for the pressure is taken from the Structural Mechanics Module: positive sign under compression, so  $p_a$  and  $p_b$  are positive parameters. [Figure 3-19](#) shows the cap in terms of the variables  $p$  and  $q$

$$q = \sqrt{3J_2(\boldsymbol{\sigma})} \text{ and } p = -I_1(\boldsymbol{\sigma})/3$$

In terms of these variables, the equation for the elliptic cap reads

$$\left(\frac{p-p_a}{p_b-p_a}\right)^2 + \left(\frac{q}{q_a}\right)^2 = 1$$

the point  $(p_a, q_a)$  in the Haigh–Westergaard coordinate system is where the elliptic cap intersects either the Mohr–Coulomb or the Drucker–Prager cone.

#### *Elliptic Cap With Hardening*

It is also possible to add isotropic hardening to the cap surface. In this case, the center of the ellipse is shifted as the volumetric plastic strain increases, also, the size of the ellipse’s semi-axes grow as hardening evolves. The intersection of the elliptic cap with the pressure axis is given by

$$p_b = p_{b0} - K_{\text{iso}} \log\left(1 + \frac{\varepsilon_{\text{pvol}}}{\varepsilon_{\text{pvol,max}}}\right)$$

here,  $p_{b0}$  is the initial value for the limit pressure  $p_b$ ,  $K_{\text{iso}}$  is the isotropic hardening modulus,  $\varepsilon_{\text{pvol}}$  the volumetric plastic strain, and  $\varepsilon_{\text{pvol,max}}$  the maximum volumetric plastic strain. Instead of providing the value for the initial pressure  $p_a$  (SI units: Pa), the ellipse’s aspect ratio  $R$  is entered. In order to have a smooth transition between the elliptic cap and the failure surface, the method described in [Ref. 3](#) is implemented.

The volumetric plastic strain  $\varepsilon_{\text{pvol}}$  is negative in compression, so the limit pressure  $p_b$  is increased from  $p_{b0}$  as hardening evolves.



See also the description of the [Drucker–Prager](#) material model in the Solid Mechanics interface documentation.

#### *The Matsuoka–Nakai Criterion*

Matsuoka and Nakai ([Ref. 5](#)) 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*. They defined the yield surface as

$$F_y = (9 + 9\mu^2)I_3 - I_1I_2 = 0$$

where the parameter  $\mu = (\tau/\sigma_n)_{\text{STP}}$  equals the maximum ratio between shear stress and normal stress in the *spatially mobilized plane* (STP-plane), and 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 in dry soils, when

$$\mu = \frac{2\sqrt{2}}{3} \tan \phi$$

and  $\phi$  denotes the angle of internal friction in the Mohr–Coulomb criterion.

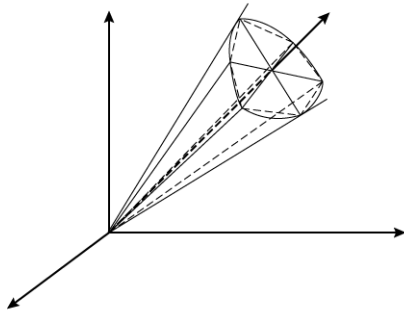


Figure 3-20: The Matsuoka–Nakai and Mohr–Coulomb criteria in the principal stress space.



See also the description of the [Matsuoka–Nakai](#) material model in the Solid Mechanics interface documentation.

#### The Lade–Duncan Criterion

The Lade–Duncan criterion was originally developed to model a large volume of laboratory sample test data of cohesionless soils. This criterion is defined as

$$F_y = kI_3 - I_1^3 = 0$$

where  $I_1$  and  $I_3$  are the first and third stress invariants, respectively, and  $k$  is a parameter related to the direction of the plastic strain increment in the triaxial plane. The parameter  $k$  can vary from 27 for hydrostatic stress conditions ( $\sigma_1 = \sigma_2 = \sigma_3$ ), up to a

critical value  $k_c$  at failure. In terms of the invariants  $I_1$ ,  $J_2$ , and  $J_3$ , this criterion can be written as

$$F_y = J_3 - \frac{1}{3}I_1J_2 + \left(\frac{1}{27} - \frac{1}{k}\right)I_1^3 = 0$$

The Lade–Duncan criterion can be fitted to the compressive meridian of the Mohr–Coulomb surface by choosing

$$k = (3 - \sin \phi)^3 / (\cos \phi^2 \cdot (1 - \sin \phi))$$

with  $\phi$  as the angle of internal friction in the Mohr–Coulomb criterion.

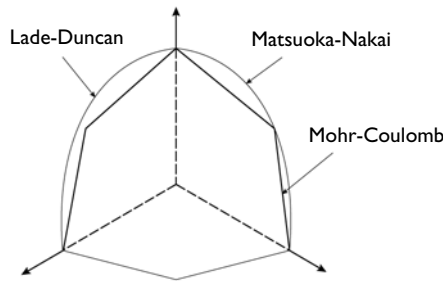


Figure 3-21: Comparing the Mohr–Coulomb, Matsuoka–Nakai, and Lade–Duncan criteria when matching the tensile meridian.



The Lade–Duncan criterion does not match the Mohr–Coulomb criterion (nor the Matsuoka–Nakai criterion) at the tensile meridian.



See also the description of the [Lade–Duncan](#) material model in the Solid Mechanics interface documentation.

### Tension Cutoff

It appears that the Mohr–Coulomb and Drucker–Prager criteria predict tensile strengths larger than the experimental measurements on soil samples. This discordance can be mended by the introduction of a *tension cutoff* criterion.

The Rankine criterion states that a material stops deforming elastically when the largest principal stress  $\sigma_1$  reaches a maximum tensile stress, also called tension cutoff limit  $\sigma_t$ .

In terms of the principal stress, the Rankine criterion reads

$$F = \sigma_1 - \sigma_t = 0$$

For soils and clays, the maximum tensile stress can be estimated from the material parameters, such as the cohesion  $c$  and the friction angle  $\phi$ . For instance, the tip of the cone in Mohr–Coulomb criterion is reached when

$$\sigma_1 = c \frac{\cos \phi}{\sin \phi}$$

therefore, the tension cutoff should be chosen such as

$$\sigma_t < c \frac{\cos \phi}{\sin \phi}$$

The Mohr–Coulomb criterion together with a tension cutoff is sometimes called the *modified Mohr–Coulomb criterion* (Ref. 21).

There might be numerical problems when implementing Rankine cutoff criterion, as the intersection of the cone in Mohr–Coulomb criterion with the tensile plane produces sharp edges in the stress space. A better approach is to use the mean stress instead of the principal stress, then the mean stress cutoff reads

$$F = p + \sigma_m = 0$$

here,  $\sigma_m$  is the maximum mean stress and  $p = -I_1/3$  is the pressure, which is negative in tension.



The Rankine tension cutoff criterion is also available with Concrete and Rocks material models.

---

### **POROUS PLASTICITY**

The modeling of plastic deformation in soils, porous metals and aggregates has a main difference with respect to traditional metal plasticity; the yield function and plastic potential are not only defined in terms of the deviatoric stress tensor (or the deviatoric stress invariant  $J_2$ ), but also include dependencies on the hydrostatic pressure.

A key concept for porous plasticity models is the evolution of the *relative density*, which is the solid volume fraction in a porous mixture. The relative density is related to the porosity (or void volume fraction)  $\phi$  by

$$\rho_{\text{rel}} = (1 - \phi)$$

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.

#### *The Shima–Oyane Criterion*

Shima and Oyane (Ref. 8) 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 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

$$Q_p(\sigma) = \left( \frac{\sigma_e}{\sigma_{ys0}} \right)^2 + \alpha(1 - \rho_{rel})^\gamma \left( \frac{p_m}{\sigma_{ys0}} \right)^2 - \rho_{rel}^m$$

here,  $\sigma_e$  is the equivalent stress,  $\sigma_{ys0}$  is the initial yield stress,  $p_m$  is the pressure, and  $\rho_{rel}$  is the relative density. The material parameters  $\alpha$ ,  $\gamma$ , 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 Gurson Criterion*

The Gurson criterion (Ref. 9) consists in a pressure dependent yield function to describe the constitutive response of porous metals, this yield function is derived from the analytical expression of an isolated void immersed in a continuum medium. The void volume fraction, or porosity  $\phi$ , is chosen as main variable. 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

$$Q_p(\sigma) = \left( \frac{\sigma_e}{\sigma_{ys0}} \right)^2 + 2\phi \cosh\left(\frac{3p_m}{2\sigma_{ys0}}\right) - (1 + \phi^2)$$

here,  $\sigma_e$  is the equivalent stress,  $\sigma_{ys0}$  is the initial yield stress,  $p_m$  is the pressure, and  $\phi$  is the porosity.

#### *The Gurson–Tvergaard–Needleman Criterion*

Tvergaard and Needleman modified [The Gurson Criterion](#) for porous plasticity to include parameters to better fit experimental data (Ref. 10–11). The resulting criterion is called the Gurson–Tvergaard–Needleman (GTN) criterion in the literature. The plastic potential for the GTN criterion reads



$$Q_p(\sigma) = \left(\frac{\sigma_e}{\sigma_{ys0}}\right)^2 + 2q_1\phi_c \cosh\left(\frac{3q_2p_m}{2\sigma_{ys0}}\right) - (1 + q_3\phi_c^2)$$

here,  $\sigma_e$  is the equivalent stress,  $\sigma_{ys0}$  is the initial yield stress,  $p_m$  is the pressure, and  $\phi_e$  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 effective void value fraction (or effective porosity)  $\phi_e$  used in the plastic potential is a function of the current porosity  $\phi$  and other material parameters:

$$\phi_e = \begin{cases} \phi & \text{for } \phi < \phi_c \\ \phi_c + \frac{\phi_m - \phi_c}{\phi_f - \phi_c}(\phi - \phi_c) & \text{for } \phi_c < \phi < \phi_f \end{cases}$$

here  $\phi_c$  is the critical void volume fraction (critical porosity) at which void coalescence begins, and  $\phi_f$  is the void volume fraction at failure. When the porosity increases up to the value of failure, the effective porosity takes a maximum value of  $\phi_m$  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  $\phi_m = 1/q_1$ .

#### *The Fleck–Kuhn–McMeeking Criterion*

The Fleck–Kuhn–McMeeking criterion (Ref. 13), also called the FKM criterion, was developed to model the plastic yielding of metal aggregates of high porosity. The yield function and associated plastic potential is derived from expressions 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

$$Q_p = \left(\frac{5}{18} \frac{\sigma_e}{p_f} + \frac{2}{3}\right)^2 + \left(\frac{5p_m}{3p_f}\right)^2 - 1$$

here,  $\sigma_e$  is the equivalent stress and  $p_m$  is the pressure. The *flow strength* of the material under hydrostatic loading,  $p_f$ , is computed from

$$p_f = 2.97(1 - \phi) \frac{2\phi_m - \phi}{\phi_m} \sigma_{ys0}$$

here,  $\sigma_{ys0}$  is the initial yield stress, and  $\phi$  is the void volume fraction (porosity). The maximum void volume fraction  $\phi_m$  typically takes the value of 36%, the limit of dense random packing of sintered powder.

#### *The FKM–GTN Criterion*

The FKM–GTN criterion is a combination of [The Fleck–Kuhn–McMeeking Criterion](#) and [The Gurson–Tvergaard–Needleman Criterion](#), intended to cover a wider range of porosities ([Ref. 14-15](#)). 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.

#### *Capped Drucker–Prager*

See the sections [The Drucker–Prager Criterion](#) and [Elliptic Cap](#) for details.

#### *Void Growth*

For the porous plasticity criteria, the change in relative density is by default computed from the change in plastic volumetric strain

$$\dot{\rho}_{rel} = -\rho_{rel} \dot{\epsilon}_{p,vol}$$

Since the relative density is related to the porosity  $\phi$  by  $\rho_{rel} = 1 - \phi$ , the change in porosity is also controlled by the change in plastic volumetric strain

$$\dot{\phi} = (1 - \phi) \dot{\epsilon}_{p,vol}$$

and the change in volumetric plastic strain  $\dot{\epsilon}_{p,vol}$  is given by the porous plasticity model.

Nucleation and shear growth are other mechanism that can trigger an increase of porosity. The increment in porosity based on growth nucleation is given by

$$\dot{\phi}_{nucleation} = \frac{f_N}{s_N \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{\epsilon_{pm} - \epsilon_N}{s_N} \right)^2} \dot{\epsilon}_{pm} \quad \text{for } p < 0$$

here,  $\epsilon_N$  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  $\epsilon_N = 0.04$ ,  $s_N = 0.1$ , and  $f_N = 0.04$ . It is assumed that nucleation appears only in tension, and that there is no nucleation in compression.

The other possible mechanism to change the porosity is the so-called void growth in shear

$$\dot{\phi}_{\text{shear}} = k_w w \phi \mathbf{n}^D : \dot{\epsilon}_{\text{pl}}$$

here,  $k_w$  is a material parameter,  $\phi$  is the current porosity,  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor, and  $\dot{\epsilon}_{\text{pl}}$  is the plastic strain rate, which depends on the porous plasticity model. The weight  $w$  is computed from stress invariants as

$$w = 1 - \frac{27J_3^2}{4J_2^3} = \sin^2(3\theta)$$

where  $\theta$  is the Lode angle. The weight variable has a value of  $w = 0$  at the compressive and tensile meridians ( $\theta = 0$  and  $\theta = \pi/3$ ), and it attains its maximum  $w = 1$  for  $\theta = \pi/6$ .

### *Isotropic Hardening*

Porous plasticity implements different kinds of isotropic hardening models to describe the hardening of the porous matrix. When applying isotropic hardening, the average flow stress

$$\sigma_{\text{fm}} = \sigma_{\text{ys0}} + \sigma_{\text{h}}(\epsilon_{\text{pm}})$$

now depends on the initial yield stress  $\sigma_{\text{ys0}}$ , the hardening function  $\sigma_{\text{h}}$ , and the equivalent plastic strain in the porous matrix  $\epsilon_{\text{pm}}$ .

Different isotropic hardening models are implemented for porous plasticity models, which are described for elastoplastic materials:

- [Perfect Plasticity](#) (no isotropic hardening)
- [Linear Isotropic Hardening](#)
- [Ludwik](#)
- [Hardening Function](#)
- [Exponential Hardening \(Elliptic Cap With Hardening\)](#)

It is also possible to include a power law relation between the equivalent plastic strain in the porous matrix  $\epsilon_{\text{pm}}$ , and the flow stress (yield level)  $\sigma_{\text{fm}}$ . For uniaxial loading, the strain stress relation is written as (ref. [Ref. 11](#))

$$\epsilon = \frac{\sigma_{\text{ys0}}}{E} \left( \frac{\sigma_{\text{fm}}}{\sigma_{\text{ys0}}} \right)^n \quad \text{for } \sigma_{\text{fm}} > \sigma_{\text{ys0}}$$

where  $\sigma_{ys0}$  is the initial yield stress,  $n$  is the hardening exponent, and the Young's modulus  $E$  is taken from the elastic material properties. By writing the onset of plasticity as  $\epsilon_0 = \sigma_{ys0}/E$ , and noting that  $\epsilon = \epsilon_0 + \epsilon_{pm}$  for  $\sigma_{fm} > \sigma_{ys0}$ , this reads

$$\epsilon_0 + \epsilon_{pm} = \epsilon_0 \left( \frac{\sigma_{fm}}{\sigma_{ys0}} \right)^n \quad \text{for } \sigma_{fm} > \sigma_{ys0}$$

which is equivalent to [Swift](#) isotropic hardening

$$\sigma_{fm}(\epsilon_{pm}) = \sigma_{ys0} \left( 1 + \frac{\epsilon_{pm}}{\epsilon_0} \right)^{1/n}$$

### ISOTROPIC HARDENING

Plasticity implements several different kinds of isotropic hardening models for elastoplastic materials:

- [Perfect Plasticity](#) (no isotropic hardening)
- [Linear Isotropic Hardening](#)
- [Ludwik](#)
- [Johnson–Cook](#)
- [Swift](#)
- [Voce](#)
- [Hockett–Sherby](#)
- [Hardening Function](#)

#### *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 on the yield surface (the regime inside the yield surface is elastic, and stress paths beyond the yield surface are not allowed).

In this case the plasticity algorithm solves either the associated or nonassociated flow rule for the plastic potential  $Q_p$

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma}$$

with the yield function

$$F_y = \phi(\sigma) - \sigma_{ys0}$$

In the settings for plasticity you specify the equivalent stress  $\phi(\sigma)$  for the yield function from von Mises stress, Tresca stress, Hill equivalent stress, or a user defined expression; and  $\sigma_{ys0}$  is the *initial yield stress* that defines the onset of plastic deformation.



When **Large plastic strain** is selected as the plasticity model for the **Plasticity** node, either the associate or nonassociated flow rule is applied as written in [Equation 3-47](#).

#### Linear Isotropic Hardening

In this case the plasticity algorithm solves either the associated or nonassociated flow rule for the plastic potential  $Q_p$

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma}$$

with the yield function

$$F_y = \phi(\sigma) - \sigma_{ys}(\epsilon_{pe})$$

where the yield stress  $\sigma_{ys}(\epsilon_{pe})$  now depends on the *equivalent plastic strain*  $\epsilon_{pe}$ .

The yield stress  $\sigma_{ys}(\epsilon_{pe})$  is then a function of the equivalent plastic strain and the *initial yield stress*  $\sigma_{ys0}$

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + E_{iso} \epsilon_{pe}$$

here, the *isotropic hardening modulus*  $E_{iso}$  is calculated 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). A value for  $E_{Tiso}$  is entered in the isotropic tangent modulus section for the Plasticity node. The Young's modulus  $E$  is taken from the parent material (Linear Elastic, Nonlinear Elastic or Hyperelastic material model). For orthotropic and anisotropic elastic materials,  $E$  represents an effective Young's modulus.

#### Ludwik

In the Ludwik model for nonlinear isotropic hardening, the yield stress  $\sigma_{ys}(\epsilon_{pe})$  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

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + k\epsilon_{pe}^n$$

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  $\sigma_{ys}(\epsilon_{pe})$  in the Johnson–Cook model is defined as a nonlinear function of the equivalent plastic strain  $\epsilon_{pe}$  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

$$\sigma_{ys}(\epsilon_{pe}) = (\sigma_{ys0} + k\epsilon_{pe}^n) \left( 1 + C \log \left( \frac{\dot{\epsilon}_{pe}}{\dot{\epsilon}_0} \right) \right) (1 - f(T_h)) \quad (3-41)$$

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 3-41](#). 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  $\dot{\epsilon}_0$  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 the melting temperature  $T_m$ , and subjected to different strain rates like 0.1/s, 1/s, and 10/s.

It should be noted that the intent of the strain rate dependent term in Equation 3-41 is to capture the hardening at high strain rates. If extrapolated to very low strain rates, a low or even negative yield stress will be predicted. Low strain rates can however be expected in some regions in a general finite element model. To resolve this, the material model as implemented, never evaluates the logarithmic term to a value smaller than zero. This means that the reference strain rate  $\dot{\epsilon}_0$  must be considered the quasistatic limit. Below this strain rate, the hardening function will be a strain rate independent, but temperature dependent, Ludwik law:

$$\sigma_{ys}(\epsilon_{pe}) = (\sigma_{ys0} + k\epsilon_{pe}^n)(1 - f(T_h))$$

#### Swift

For nonsaturating materials, the Swift power-law equation relates the initial yield stress  $\sigma_{ys0}$  and the isotropic hardening  $\sigma_h$ , to the equivalent plastic strain as

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + \sigma_h(\epsilon_{pe}) = k(\epsilon_0 + \epsilon_{pe})^n$$

here,  $k$  is the strength coefficient,  $n$  is the hardening exponent, and  $\epsilon_0$  is a reference strain. Noting that at zero plastic strain the initial yield stress is related to the strength coefficient and hardening exponent as

$$\sigma_{ys}(0) = \sigma_{ys0} = k\epsilon_0^n$$

the yield stress is written as

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} \left(1 + \frac{\epsilon_{pe}}{\epsilon_0}\right)^n$$

#### Voce

The Voce rule for nonlinear isotropic hardening is intended for materials that exhibit a saturating evolution of hardening. The isotropic hardening  $\sigma_h$  is exponentially related to the equivalent plastic strain as

$$\sigma_h(\epsilon_{pe}) = \sigma_{sat} \left(1 - e^{-\beta\epsilon_{pe}}\right)$$

The yield stress  $\sigma_{ys}(\epsilon_{pe})$  is then defined as

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + \sigma_h(\epsilon_{pe}) = \sigma_{ys0} + \sigma_{sat} \left(1 - e^{-\beta\epsilon_{pe}}\right)$$

The value of the *saturation exponent* parameter  $\beta$  determines the saturation rate of the hysteresis loop for cyclic loading. The *saturation flow stress*  $\sigma_{\text{sat}}$  characterizes the maximum distance by which the yield surface can expand in the stress space. For values  $\epsilon_{\text{pe}} \gg 1/\beta$ , the yield stress saturates to

$$\sigma_{\text{ys}} \rightarrow \sigma_{\text{ys}0} + \sigma_{\text{sat}}$$

#### 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

$$\sigma_{\text{ys}}(\epsilon_{\text{pe}}) = \sigma_{\text{ys}0} + (\sigma_{\infty} - \sigma_{\text{ys}0}) \left( 1 - e^{-m \epsilon_{\text{pe}}^n} \right)$$

where  $\sigma_{\infty}$  is the *steady-state flow stress*,  $m$  the saturation coefficient and  $n$  the saturation exponent. For values  $m \epsilon_{\text{pe}}^n \gg 1$  the yield stress saturates to

$$\sigma_{\text{ys}} \rightarrow \sigma_{\infty}$$

#### 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  $\sigma_{\text{h}}(\epsilon_{\text{pe}})$  such that the yields stress reads

$$\sigma_{\text{ys}}(\epsilon_{\text{pe}}) = \sigma_{\text{ys}0} + \sigma_{\text{h}}(\epsilon_{\text{pe}})$$



The internal variable for the equivalent plastic strain is named `solid.epe`. The equivalent plastic strain evaluated at Gauss points is named `solid.epeGp`.

---



When **Large plastic strain** is selected as the plasticity model for the Plasticity node, either the associate or nonassociated flow rule is applied as written in [Equation 3-47](#).

---



## KINEMATIC HARDENING

There are few options for computing either linear or nonlinear kinematic hardening for plasticity:

- No kinematic hardening (default)
- Linear kinematic hardening
- Armstrong–Frederik
- Chaboche

For any of the kinematic hardening models, the algorithm solves either the associated or nonassociated flow rule for the plastic potential  $Q_p$

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma}$$

with the yield function defined as

$$F_y = \phi(\sigma - \sigma_{\text{back}}) - \sigma_{ys}$$

Here,  $\sigma_{ys}$  is the yield stress (which may include a linear or nonlinear isotropic hardening model), and the equivalent stress  $\phi(\sigma)$  is either the von Mises, Tresca, or Hill stress; or a user defined expression. The stress tensor used in the yield function is shifted by what is usually called the *back stress*,  $\sigma_{\text{back}}$ .

### *Linear Kinematic Hardening*

The back stress is generally not only a function of the current plastic strain but also of its history. In the case of linear kinematic hardening, the back stress  $\sigma_{\text{back}}$  is a linear function of the plastic strain tensor  $\epsilon_p$ , this is also known as *Prager's hardening rule*.

The implementation of linear kinematic hardening assumes a linear evolution of the back stress tensor with respect to the plastic strain tensor:

$$\sigma_{\text{back}} = \frac{2}{3} C_k \epsilon_p$$

where the *kinematic hardening modulus*  $C_k$  is calculated from

$$\frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

The value for  $E_k$  is entered in the *kinematic tangent modulus* section and the Young's modulus  $E$  is taken from the linear or nonlinear elastic material model. For orthotropic

and anisotropic elastic materials,  $E$  represents an averaged Young's modulus. Note that some authors define the kinematic hardening modulus as  $H_k = 2/3C_k$ .



When kinematic hardening is added, both the plastic potential and the yield surface are calculated with effective invariants, that is, the invariants of the tensor defined by the difference between the stress tensor minus the back-stress,  $\sigma_{\text{eff}} = \sigma - \sigma_{\text{back}}$ . The invariant of effective deviatoric tensor is named `solid.II2SEff`, which is used when a von Mises, Tresca or Hill orthotropic plasticity is computed together with kinematic hardening.

#### *Armstrong–Frederik Hardening Model*

Armstrong and Frederick (Ref. 1) added memory to Prager's linear kinematic hardening model. This nonlinear kinematic hardening model makes it possible to capture the Bauschinger effect and nonlinear behavior by nonsymmetric tension-compression loading.

The nonlinear evolution of the back stress  $\sigma_{\text{back}}$  is governed by the rate

$$\dot{\sigma}_{\text{back}} = \frac{2}{3}C_k \dot{\epsilon}_p - \gamma_k \dot{\epsilon}_{pe} \sigma_{\text{back}}$$

here,  $C_k$  is the kinematic hardening modulus,  $\gamma_k$  is a kinematic hardening parameter, and  $\epsilon_{pe}$  the equivalent plastic strain. Setting  $\gamma_k = 0$  recovers Prager's rule for linear kinematic hardening.

To solve this rate, internal degrees of freedom are added to account for the back stress components. In order to have the same units as used for the plastic strain, the algorithm solves for the back strain  $\epsilon_{\text{back}}$  as proposed in (Ref. 2), which is related to the back stress as

$$\sigma_{\text{back}} = \frac{2}{3}C_k \epsilon_{\text{back}}$$

The nonlinear evolution for the back strain reads

$$\dot{\epsilon}_{\text{back}} = \dot{\epsilon}_p - \gamma_k \dot{\epsilon}_{pe} \epsilon_{\text{back}}$$

#### *Chaboche Hardening Model*

Chaboche (Ref. 2) proposed a nonlinear kinematic hardening model based on the superposition of  $N$  back stress tensors

$$\sigma_{\text{back}} = \sum_i^N \sigma_{\text{back},i}$$

each of these back stress tensors  $\sigma_{\text{back},i}$  follows a nonlinear Frederick–Armstrong kinematic hardening rule

$$\dot{\sigma}_{\text{back},i} = \frac{2}{3}C_i\dot{\varepsilon}_p - \gamma_i\dot{\varepsilon}_p\sigma_{\text{back},i}$$

Practitioners would normally select  $\gamma_k = 0$  for one of the back stress equations, thus recovering Prager’s linear rule for that branch

$$\sigma_{\text{back},0} = \frac{2}{3}C_0\varepsilon_p$$

The back stress tensor  $\sigma_{\text{back}}$  is then defined by the superposition of  $N$  back stress tensors

$$\sigma_{\text{back}} = \frac{2}{3}C_0\varepsilon_p + \sum_{i=1}^N \sigma_{\text{back},i}$$

As done for Armstrong–Frederik kinematic hardening, the algorithm solves for the back strain tensors  $\varepsilon_{\text{back},i}$  instead of the back stress tensors. The change of variables is

$$\sigma_{\text{back},i} = \frac{2}{3}C_i\varepsilon_{\text{back},i}$$

and the nonlinear evolution for the back strain tensors reads

$$\dot{\varepsilon}_{\text{back},i} = \dot{\varepsilon}_p - \gamma_i\dot{\varepsilon}_p\varepsilon_{\text{back},i}$$

## INTRODUCTION TO SMALL AND LARGE PLASTIC STRAINS

There are two implementations of plasticity available in COMSOL Multiphysics. One is based on the additive decomposition of strains, which is the most suitable approach in the case of small strains, and the other one is based on the multiplicative decomposition of the deformation gradient, which is more suitable when large plastic strains occurs. The additive and multiplicative decomposition of strains is described in [Inelastic Strain Contributions](#).

When *small plastic strain* is selected as the plasticity model, an additive decomposition is used. If the elastic or plastic strains are large, the additive

decomposition might produce incorrect results. As an example, the volume preservation, which is an important assumption in metal plasticity, will no longer be respected. The additive decomposition of strains is however widely used both for metal and soil plasticity.



When **Small plastic strain** is selected as the plasticity model for the **Plasticity** node, and the **Include geometric nonlinearity** check box is selected on the study **Settings** window, a Cauchy stress tensor is used to evaluate the yield function and plastic potential. The components of this stress tensor are oriented along the material directions, so it can be viewed as a scaled second Piola–Kirchhoff stress tensor. The additive decomposition of strains is understood as the summation of Green–Lagrange strains.

When *large plastic strain* is selected as the plasticity model, the total deformation gradient tensor is multiplicatively decomposed into an elastic deformation gradient and a plastic deformation gradient.

#### PLASTIC FLOW FOR SMALL STRAINS

The *flow rule* defines the relationship between the increment of the plastic strain tensor  $\dot{\epsilon}_p$  and the current state of stress,  $\sigma$ , for a yielded material subject to further loading. When **Small plastic strain** is selected as the plasticity model for the **Plasticity** node, the direction of the plastic strain increment is defined by

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma}$$

Here,  $\lambda$  is a positive multiplier (also called the *consistency parameter* or *plastic multiplier*), which depends on the current state of stress and the load history, and  $Q_p$  is the *plastic potential*.



The “dot” (for  $\dot{\epsilon}_p$ ) means the rate at which the plastic strain tensor changes with respect to  $\partial Q_p / \partial \sigma$ . It does not represent a true time derivative. Some authors call this formulation *rate independent plasticity*.

The direction of the plastic strain increment  $\dot{\epsilon}_p$  is perpendicular to the surface (in the hyperspace spanned by the stress tensor components) defined by the plastic potential  $Q_p$ .

The plastic multiplier  $\lambda$  is determined by the *complementarity* or *Kuhn–Tucker conditions*

$$\lambda \geq 0, F_y \leq 0 \text{ and } \lambda F_y = 0 \quad (3-42)$$

where  $F_y$  is the *yield function*. The yield surface encloses the elastic region defined by  $F_y < 0$ . Plastic flow occurs when  $F_y = 0$ .

If the plastic potential and the yield surface coincide with each other ( $Q_p = F_y$ ), the flow rule is called *associated*, and the rate in Equation 3-43 is solved together with the conditions in Equation 3-42.

$$\dot{\epsilon}_p = \lambda \frac{\partial F_y}{\partial \sigma} \quad (3-43)$$

For a *nonassociated* flow rule, the yield function does not coincide with the plastic potential, and together with the conditions in Equation 3-42, the rate in Equation 3-44 is solved for the plastic potential  $Q_p$  (often, a smoothed version of  $F_y$ ).

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma} \quad (3-44)$$

The evolution of the plastic strain tensor  $\dot{\epsilon}_p$  (with either Equation 3-43 or Equation 3-44, plus the conditions in Equation 3-42) is implemented at Gauss points in the plastic element `e1plastic`.

### PLASTIC FLOW FOR LARGE STRAINS

When **Large plastic strain** is selected as the plasticity model for the Plasticity node, a multiplicative decomposition of deformation (Ref. 3, Ref. 4, and Ref. 5) is used, and the associated plastic flow rule can be written as the *Lie derivative* of the elastic left Cauchy–Green deformation tensor  $B_{el}$ :

$$-\frac{1}{2}L(B_{el}) = \lambda \frac{\partial \Phi}{\partial \tau} B_{el} \quad (3-45)$$

The plastic multiplier  $\lambda$  and the yield function  $\Phi$  (written in terms of the Kirchhoff stress tensor  $\tau$ ) satisfy the Kuhn–Tucker condition, as done for infinitesimal strain plasticity

$$\lambda \geq 0, \Phi \leq 0 \text{ and } \lambda\Phi = 0$$



The yield function  $\Phi$  in Ref. 3 and Ref. 4 was written in terms of Kirchhoff stress  $\tau$  and not Cauchy stress  $\sigma$  because the authors defined the plastic dissipation with the conjugate energy pair  $\tau$  and  $\mathbf{d}$ , where  $\mathbf{d}$  is the *rate of strain tensor*.

The Lie derivative of  $B_{el}$  is then written in terms of the plastic right Cauchy–Green rate

$$\mathcal{L}(B_{el}) = FC_p^{-1}F^T \quad (3-46)$$

By using Equation 3-45 and Equation 3-46, the either associated or nonassociated plastic flow rule for large strains is written as (Ref. 4)

$$-\frac{1}{2}FC_p^{-1}F^T = \lambda \frac{\partial Q_p}{\partial \tau} B_{el} \quad (3-47)$$

together with the Kuhn–Tucker conditions for the plastic multiplier  $\lambda$  and the yield function  $F_y$

$$\lambda \geq 0, F_y \leq 0 \text{ and } \lambda F_y = 0 \quad (3-48)$$

For the associated flow rule, the plastic potential and the yield surface coincide with each other ( $Q_p = F_y$ ), and for the nonassociated case, the yield function does not coincide with the plastic potential.

In COMSOL Multiphysics, the elastic left Cauchy–Green tensor is written in terms of the deformation gradient and the right Cauchy–Green tensor, so  $B_{el} = FC_p^{-1}F^T$ . The flow rule then reads

$$C_p^{-1} = -2\lambda F^{-1} \frac{\partial Q_p}{\partial \tau} FC_p^{-1} \quad (3-49)$$

The plastic flow rule is then solved at Gauss points in the plastic element `elplastic` for the inverse of the plastic deformation gradient  $F_p^{-1}$ , so that the variables in Equation 3-49 are replaced by

$$C_p^{-1} = F_p^{-1}F_p^{-T}, C_p^{-1} = F_p^{-1}F_p^{-T} + F_p^{-1}F_p^{-T} \text{ and } B_{el} = FF_p^{-1}F_p^{-T}F^T$$

After integrating the flow rule in Equation 3-49, the plastic Green–Lagrange strain tensor is computed from the plastic deformation tensor

$$\varepsilon_p = \frac{1}{2}(F_p^T F_p - I)$$

and the elastic Green–Lagrange strain tensor is computed from the elastic deformation gradient tensor  $F_{el} = F F_p^{-1}$

$$\varepsilon_{el} = \frac{1}{2}(F_{el}^T F_{el} - I)$$



When **Large plastic strain** is selected as the plasticity model for the Plasticity node, the equivalent plastic strain variable is computed as the *true* equivalent plastic strain (also called *Hencky* or *logarithmic* plastic strain).



When either **Large plastic strain** or **Small plastic strain** is selected as the plasticity model for the Plasticity node, the out-of-plane shear strain components are not computed in 2D, neither for plane stress nor plane strain assumption.

### NUMERICAL SOLUTION OF THE ELASTOPLASTIC CONDITIONS

A backward Euler discretization of the pseudo-time derivative is used in the plastic flow rule. For small plastic strains, this gives

$$\varepsilon_p - \varepsilon_{p, \text{old}} = \Lambda \frac{\partial Q_p}{\partial \sigma}$$

where “old” denotes the previous time step and  $\Lambda = \lambda \Delta t$ , where  $\Delta t$  is the pseudo-time step length.

For large plastic strains, [Equation 3-49](#) is numerically solved with the so-called *exponential mapping* technique

$$C_p^{-1} = F^{-1} \exp\left(-2\Lambda \frac{\partial Q_p}{\partial \sigma}\right) F C_{p, \text{old}}^{-1}$$

where  $C_p^{-1} = F_p^{-1} F_p^{-T}$  and  $C_{p, \text{old}}^{-1} = F_{p, \text{old}}^{-1} F_{p, \text{old}}^{-T}$ .

For each Gauss point, the plastic state variables ( $\varepsilon_p$  or  $F_p^{-1}$ , depending on whether small strain or large strain plasticity is selected) and the plastic multiplier,  $\Lambda$ , are computed by solving either of the above time-discretized flow rules together with the complementarity conditions

$$\Lambda \geq 0, \quad F_y \leq 0, \quad \Lambda F_y = 0$$

This is done as follows (Ref. 3):

- 1 Elastic-predictor: Try the elastic solution  $\varepsilon_p = \varepsilon_{p,old}$  (or  $F_p^{-1} = F_{p,old}^{-1}$ ) and  $\Lambda = 0$ . If this satisfies  $F_y \leq 0$  it is done.
- 2 Plastic-corrector: If the elastic solution does not work (this is  $F_y > 0$ ), solve the nonlinear system consisting of the flow rule and the equation  $F_y = 0$  using a damped Newton method.



The numerical tolerance to fulfill the condition  $F_y = 0$  is given in SI units of Pascals, and it depends on the initial yield stress (in case of plasticity and porous plasticity) or it is defined in terms of other material parameter (for soil plasticity). This numerical tolerance is 0.1% the value defined in the variable `item.tol.`, where `item` is the name of the node.

### ENERGY DISSIPATION

Since plasticity is an inelastic process, the dissipated energy density can be calculated by integrating the pseudo-rate given by

$$\dot{W}_p = \sigma : \dot{\varepsilon}_p = \lambda \sigma : \frac{\partial Q_p}{\partial \sigma}$$

As plasticity is rate independent, the plastic dissipation density  $W_p$  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 a volume integration of the plastic dissipation density  $W_p$ .



When the **Calculate dissipated energy** check box is selected, the plastic dissipation density is available under the variable `solid.Wp` and the total plastic dissipation under the variable `solid.Wp_tot`.

### *Failure Criteria for Concrete, Rocks, and Other Brittle Materials*

In this section:

- [The Bresler–Pister Criterion](#)
- [The Willam–Warnke Criterion](#)



- The Ottosen Criterion
- The Original Hoek–Brown Criterion
- The Generalized Hoek–Brown Criterion

#### THE BRESLER–PISTER CRITERION

The Bresler–Pister criterion (Ref. 4, Ref. 19) was originally devised to predict the strength of concrete under multiaxial stresses. This failure criterion is an extension of [The Drucker–Prager Criterion](#) to brittle materials, and it can be expressed in terms of the stress invariants as

$$F_y = \sqrt{J_2} + k_1 I_1^2 + k_2 I_1 + k_3 \quad (3-50)$$

here,  $k_1$ ,  $k_2$ , and  $k_3$  are parameters obtained from the uniaxial compressive strength  $\sigma_c$ , the uniaxial tensile strength  $\sigma_t$ , and the biaxial compressive strength  $\sigma_b$

$$k_1 = \frac{\sigma_b \sigma_c + 2\sigma_c \sigma_t - 3\sigma_b \sigma_t}{\sqrt{3}(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$

$$k_2 = \frac{(\sigma_t - \sigma_c)(\sigma_b(\sigma_c + \sigma_t - 4\sigma_b) - \sigma_c \sigma_t)}{\sqrt{3}(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$

$$k_3 = \frac{-\sigma_c \sigma_b \sigma_t (\sigma_t + 8\sigma_b - 3\sigma_c)}{\sqrt{3}(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$

All the strengths are considered with a positive sign. Note that for typical strength values, the parameters  $k_1$  and  $k_2$  are positive while  $k_3$  is negative.

#### THE WILLAM–WARNKE CRITERION

The Willam–Warnke criterion (Ref. 12) 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 was defined as

$$F_y = \sqrt{\frac{3}{5}} \frac{\tau_{\text{oct}}}{\sigma_c} + r(\theta) \left( \left( \frac{1}{\sigma_t} - \frac{1}{\sigma_b} \right) \sigma_{\text{oct}} - 1 \right) \quad (3-51)$$

where  $\sigma_c$  is the uniaxial compressive strength,  $\sigma_t$  is uniaxial tensile strength, and  $\sigma_b$  is the biaxial compressive strength (all parameters are positive). The octahedral normal and shear stresses are defined as usual; see [Other Stress Invariants](#)

$$\sigma_{\text{oct}} = I_1/3, \text{ and } \tau_{\text{oct}} = \sqrt{2J_2/3}$$

so the criterion in [Equation 3-51](#) can be written in term of stress invariants as

$$F_y = \sqrt{J_2} + \sqrt{\frac{5}{2}} r(\theta) \sigma_c \left( \left( \frac{1}{\sigma_t} - \frac{1}{\sigma_b} \right) \frac{I_1}{3} - 1 \right)$$

The dimensionless function  $r(\theta)$  describes the segment of an ellipse on the octahedral plane as a function of the Lode angle  $\theta$

$$r(\theta) = \frac{2r_c(r_c^2 - r_t^2) \cos \theta + r_c(2r_t - r_c) \sqrt{4(r_c^2 - r_t^2) \cos^2 \theta + 5r_t^2 - 4r_c r_t}}{4(r_c^2 - r_t^2) \cos^2 \theta + (r_c - 2r_t)^2}$$

Here, the tensile and compressive meridian  $r_t$  and  $r_c$  are defined in terms of the strengths  $\sigma_c$ ,  $\sigma_b$ , and  $\sigma_t$ :

$$r_t = \sqrt{\frac{6}{5}} \frac{1}{\sigma_c} \frac{\sigma_b \sigma_t}{2\sigma_b + \sigma_t}$$

$$r_c = \sqrt{\frac{6}{5}} \frac{\sigma_b \sigma_t}{3\sigma_b \sigma_t + \sigma_b \sigma_c - \sigma_t \sigma_c}$$

The function  $r(\theta)$  can be interpreted as the friction angle which depends on the Lode angle  $\theta$  (Ref. 12).

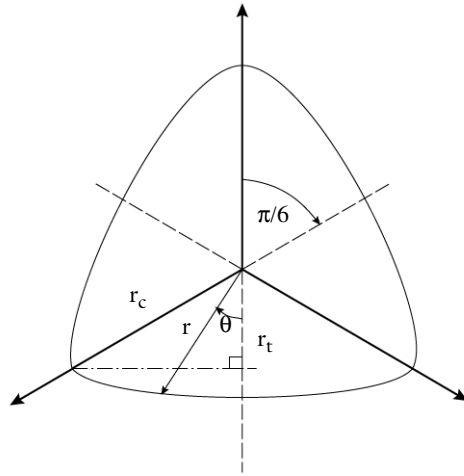


Figure 3-22: The deviatoric section of the Willam–Warnke failure criterion.

#### THE 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 curve in the pi-plane changes from 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. 20).

The Ottosen criterion is commonly written as (Ref. 19, Ref. 20):

$$F_y = \frac{a}{\sigma_c} J_2 + \lambda(\theta) \sqrt{J_2} + b I_1 - \sigma_c$$

Here,  $\sigma_c > 0$  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  $\theta$  and the dimensionless parameters  $k_1 > 0$  and  $k_2 > 0$ .

$$\lambda(\theta) = \begin{cases} k_1 \cos\left(\frac{1}{3} \arccos(k_2 \cos(3\theta))\right) & J_3 > 0 \\ k_1 \cos\left(\frac{\pi}{3} - \frac{1}{3} \arccos(-k_2 \cos(3\theta))\right) & J_3 < 0. \end{cases}$$

The parameter  $k_1$  is called the *size factor*. The parameter  $k_2$  (also called *shape factor*) is positive and bounded to  $0 \leq k_2 \leq 1$  (Ref. 19, Ref. 20).

Typical values for these parameters are obtained by curve-fitting the uniaxial compressive strength  $\sigma_c$ , uniaxial tensile strength  $\sigma_t$ , 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  $\sigma_c$ ,  $\sigma_b$ , and  $\sigma_t$  are positive.

TABLE 3-3: TYPICAL PARAMETER VALUES FOR OTTOSEN FAILURE CRITERION (Ref. 20).

$\sigma_t/\sigma_c$	$a$	$b$	$k_1$	$k_2$	$\lambda_t$	$\lambda_c$
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  $\alpha = 0$  and  $\lambda$  independent of the Lode angle.

### THE 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. 7)

$$\sigma_1 = \sigma_3 + \sqrt{m\sigma_c\sigma_3 + s\sigma_c^2}$$

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*),  $\sigma_c$  is the uniaxial compressive strength of the intact rock (positive parameter), 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

$$\sigma_1 = \sigma_3 + \sqrt{-m\sigma_c\sigma_1 + s\sigma_c^2}$$

with  $\sigma_c$ ,  $m$ , and  $s$  positive material parameters. (In this case, note that  $\sigma_1 < s\sigma_c/m$ ).

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. 9).

TABLE 3-4: 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  $I_1$  and  $J_2$  and the Lode angle  $0 \leq \theta \leq \pi/3$ , so

$$F_y = 2\sqrt{J_2}\sin\left(\theta + \frac{\pi}{3}\right) - \sigma_c\sqrt{s - m\frac{\sigma_1}{\sigma_c}} = 0$$

### THE GENERALIZED HOEK–BROWN CRITERION

The generalized Hoek–Brown criterion was developed in order to fit the Geological Strength Index (GSI) classification of isotropic rock masses (Ref. 8). 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

$$\sigma_1 - \sigma_3 = \sigma_c \left( s - m_b \frac{\sigma_1}{\sigma_c} \right)^a$$

In terms of the invariants  $J_2$  and the Lode angle  $0 \leq \theta \leq \pi/3$  this equals

$$F_y = 2\sqrt{J_2} \sin\left(\theta + \frac{\pi}{3}\right) - \sigma_c \left( s - m_b \frac{\sigma_1}{\sigma_c} \right)^a = 0$$

where  $\sigma_1 \geq \sigma_2 \geq \sigma_3$  are the principal stresses (using the Structural Mechanics Module conventions) of the *effective stress tensor* (this is, the stress tensor minus the fluid pore pressure).

The positive parameter  $m_b$  is a reduced value of the material constant  $m_i$ :

$$m_b = m_i \exp\left(\frac{\text{GSI} - 100}{28 - 14D}\right)$$

$s$  and  $a$  are positive parameters for the rock mass given by the following relationships:

$$s = \exp\left(\frac{\text{GSI} - 100}{9 - 3D}\right)$$

$$a = \frac{1}{2} + \frac{1}{6} \left( \exp\left(\frac{-\text{GSI}}{15}\right) - \exp\left(\frac{-20}{3}\right) \right)$$

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 3-5: 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

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 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 model* (MCC) is the most commonly used model due to its smooth yield surface, and it is the one implemented in the Geomechanics Module.

The modified Cam-Clay model is a so-called critical state model, where the loading and unloading of the material follow different trajectories in stress space. The model also features hardening and softening of clays. Different formulations can be found in textbooks, see for instance [Ref. 15](#), [Ref. 16](#), and [Ref. 17](#).

The yield function is written in terms of the stress invariants

$$\begin{aligned}q &= \sqrt{3J_2(\boldsymbol{\sigma})} \\ p &= -I_1(\boldsymbol{\sigma})/3\end{aligned}$$

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  $pq$ -plane, with a cross section independent of the Lode angle. Note that  $p$ ,  $q$ , and  $p_c$  are positive variables.

The parameter  $M > 0$  defines the slope of the *critical state line* in the  $pq$ -plane. This parameter can be related to the angle of internal friction  $\phi$  in the Mohr–Coulomb criterion as

$$M = \frac{6 \sin(\phi)}{3 - \sin(\phi)} \quad (3-52)$$

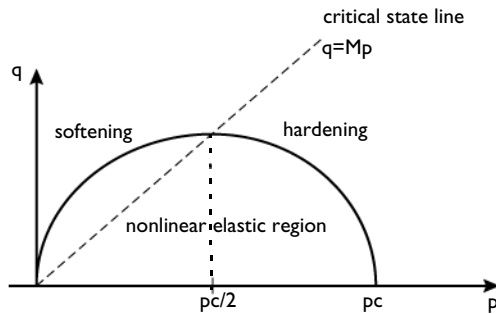


Figure 3-23: Modified Cam-Clay ellipse in the  $pq$ -plane. The ellipse circumscribes a nonlinear elastic region.



The slope of the *critical state line*  $M$  can either be a material property or it can be derived from the angle of internal friction  $\phi$ .

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 3-24](#).



The void ratio  $e$  is the ratio between the pore space and solid volume. It can be written in terms of the porosity  $\varepsilon$  as  $e = \varepsilon / (1 - \varepsilon)$ .



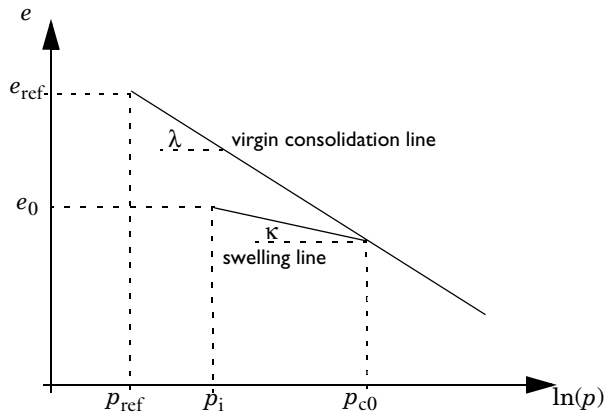


Figure 3-24: Slopes of the virgin isotropic consolidation line and swelling line in the  $e$  versus  $\ln(p)$  plane.

The compression index  $\lambda$  is the slope of the *virgin isotropic consolidation line*, and the swelling index  $\kappa$  is the slope of the *swelling line* (also called unloading-reloading line) in the  $e$  versus  $\ln(p)$  plane. The reference void ratio  $e_{\text{ref}}$  is measured at the reference pressure  $p_{\text{ref}}$  and initial void ratio  $e_0$  is measured at the initial mean stress  $p_i$ . The starting value of initial mean stress  $p_i$  is the reference pressure  $p_{\text{ref}}$ . The equation for the virgin isotropic consolidation line is written as

$$e = e_{\text{ref}} - \lambda \ln\left(\frac{p}{p_{\text{ref}}}\right)$$

The void ratio at the initial consolidation 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

$$e = e_0 - \kappa \ln\left(\frac{p}{p_i}\right)$$

where  $e_0$  is the initial void ratio calculated as

$$e_0 = e_{c0} + \kappa \ln\left(\frac{p_{c0}}{p_i}\right)$$



The initial void ratio  $e_0$  can be directly entered as user input. In this case, the void ratio at 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

$$\varepsilon_{\text{vol}} = \frac{V - V_0}{V_0} = \frac{e - e_0}{1 + e_0}$$

For elastic response, the incremental volumetric elastic strain is written as

$$d\varepsilon_{\text{el,vol}} = \frac{de}{1 + e_0} = -\kappa \frac{dp}{p} \quad (3-53)$$

Integrating Equation 3-53, the equation is reformulated as

$$(1 + e_0)\varepsilon_{\text{el,vol}} = -\kappa \ln\left(\frac{p}{p_i}\right)$$

Then the nonlinear relation between pressure and volumetric strain is obtained by

$$p = p_i e^{-B_{\text{el}}\varepsilon_{\text{el,vol}}} \quad (3-54)$$

where

$$B_{\text{el}} = \frac{1 + e_0}{\kappa}$$



The contribution to initial mean stress  $p_i$  also comes from the first invariant of the initial or external stress tensor of **Initial Stress and Strain** or **External Stress** feature. 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 **Initial Stress and Strain** or **External Stress** features the initial mean stress is nonzero and equal to the reference pressure  $p_{\text{ref}}$ . The reference pressure acts as an in situ stress.

$$p_i = p_{\text{ref}} \quad (3-55)$$

The tangent bulk modulus  $K_t$  is written as

$$K_t = \frac{1 + e_0}{\kappa} p$$

The secant bulk modulus  $K_s$  (or bulk modulus  $K$ ) is written as

$$K_s = \frac{p_i - p}{\varepsilon_{\text{el,vol}}}$$



When a constant Poisson's ratio is specified in the Modified Cam-Clay model, the shear modulus is calculated from the secant bulk modulus and Poisson's ratio, which in turn gives a variable shear modulus.

The total volumetric strain increment is written as

$$d\varepsilon_{\text{vol}} = \frac{de}{1 + e_0} = -\frac{\kappa}{1 + e_0} \frac{dp}{p} - \frac{\lambda - \kappa}{1 + e_0} \frac{dp_c}{p_c}$$

The evolution of void ratio is then written as

$$e = (1 + e_0)\varepsilon_{\text{vol}} + e_0 \quad \text{or} \quad (3-56)$$

$$e = e_{\text{ref}} - \kappa \ln\left(\frac{p}{p_1}\right) - \lambda \ln\left(\frac{p_c}{p_{\text{ref}}}\right) + \kappa \ln\left(\frac{p_c}{p_1}\right) \quad (3-57)$$

Equation 3-56 is used to compute the void ratio which matches exactly with Equation 3-57 which may 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  $\varepsilon_{\text{p,vol}}$  as

$$\dot{p}_c = -\frac{1+e_0}{\lambda-\kappa} p_c \dot{\varepsilon}_{p,\text{vol}} \quad (3-58)$$



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*  $\kappa$ , and the *compression index*  $\lambda$ , which are positive parameters that fulfill

$$0 < \kappa < \lambda \quad \text{and} \quad e_{\text{ref}} > 0$$



If an **Initial Stress and Strain** node is added to the Cam-Clay material, the initial consolidation pressure  $p_{c0}$  must be equal or larger than one third of minus the trace of the initial stress tensor, otherwise the initial stress state is outside the Cam-Clay ellipse.

#### Volumetric Elastic Deformation

The stress-strain relation beyond the elastic range is of great importance in soil mechanics. For additive decomposition of strains, Cauchy's stress tensor is written as

$$\sigma = \text{dev}(\sigma_0) + p_i e^{-B_{\text{el}} \varepsilon_{\text{el},\text{vol}}} + 2G \text{dev}(\varepsilon_{\text{el}})$$

with

$$B_{\text{el}} = \frac{1+e_0}{\kappa}$$

Here,  $\sigma$  is the stress tensor,  $\varepsilon$  is the total strain tensor,  $\varepsilon_{\text{el}}$  is the elastic strain tensor,  $\sigma_0$  is the initial or external stress 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 (see [Equation 3-54](#) and [Equation 3-55](#)). This is needed as the MCC model do not have stiffness at zero stress. The reference pressure appears as an additional term in the variational formulation (weak equation).

As opposed to the [Linear Elastic Material](#), the MCC model introduces a nonlinear relation for the hydrostatic pressure as a function of the volumetric elastic strain given by [Equation 3-54](#).

#### Hardening and Softening

The yield surface for the modified Cam-Clay model reads

$$F_y = q^2 + M^2(p - p_c)p$$

The yield surface and the associated flow rule ( $Q_p = F_y$ ) give the rate equation for the plastic strains. The plastic strain tensor  $\epsilon_p$  is calculated from the plastic multiplier  $\lambda_p$  and the derivatives of the plastic potential  $Q_p$  with respect to the stress tensor  $\sigma$

$$\dot{\epsilon}_p = \lambda_p \frac{\partial Q_p}{\partial \sigma} = \lambda_p \left( \frac{\partial Q_p}{\partial I_1} \frac{\partial I_1}{\partial \sigma} + \frac{\partial Q_p}{\partial J_2} \frac{\partial J_2}{\partial \sigma} \right)$$



Here,  $\lambda_p$  stands for the plastic multiplier, see [Plastic Flow for Small Strains](#) and [Isotropic Hardening](#).

The plastic strain rate tensor  $\dot{\epsilon}_p$  includes both deviatoric and isotropic parts. Note that

$$\partial I_1 / \partial \sigma = \mathbf{I}$$

and

$$\partial J_2 / \partial \sigma = \text{dev}(\sigma)$$

These relations can be used for writing the plastic flow as

$$\dot{\epsilon}_p = \lambda_p \left( -\frac{1}{3} \frac{\partial Q_p}{\partial p} \mathbf{I} + \frac{\partial Q_p}{\partial q} \frac{3}{2q} \text{dev}(\sigma) \right)$$

since the associated flow rule implies a plastic potential such as

$$Q_p = F_y = q^2 + M^2(p - p_c)p$$

the plastic flow rule simplifies to

$$\dot{\epsilon}_p = \lambda_p \left( -\frac{1}{3} M^2 (2p - p_c) \mathbf{I} + 3 \text{dev}(\sigma) \right)$$

The trace of the plastic strain rate tensor (the *volumetric plastic strain rate*  $\dot{\epsilon}_{\text{pvol}}$ ) then reads

$$\dot{\epsilon}_{\text{pvol}} = \text{trace}(\dot{\epsilon}_{\text{p}}) = \lambda_{\text{p}} M^2 (p_{\text{c}} - 2p)$$

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 *J2* 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 the consolidation pressure  $p_{\text{c}}$  as a function of volumetric plastic strain, as described in [Equation 3-58](#).

Hardening introduces changes in the shape of the Cam-Clay ellipse, since its major semiaxis depends on the value of the consolidation pressure  $p_{\text{c}}$ . The *initial consolidation pressure*  $p_{\text{c}0}$  defines the size of the ellipse before plastic deformation occurs.

#### *Including pore pressure*

When an external pore pressure  $p_{\text{f}}$  is added to the MCC material, the yield function is shifted along the  $p$  axis, and the yield function reads:

$$F_{\text{y}} = q^2 + M^2 (p - p_{\text{f}} - p_{\text{c}})(p - p_{\text{f}})$$

The quantity  $p - p_{\text{f}}$  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  $p_{\text{f}}$  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 model (SCC) was developed ([Ref. 22](#), [Ref. 23](#)) 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 model (MSCC) was further developed to model destructured, naturally structured and artificially structured clays (Ref. 22), and it is the one implemented in the Geomechanics Module.

In the MSCC model, the reduction of mean effective stress due to structure degradation,  $p_b$ , depends on the shear plastic strain  $\varepsilon_{p,dev}$ . 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

$$p_b = \begin{cases} p_{bi}e^{-\varepsilon_{p,dev}} & \text{if } \varepsilon_{p,dev} \leq \varepsilon_{p,deve} \\ p_{bf}e^{-d_s(\varepsilon_{p,dev} - \varepsilon_{p,deve})} & \text{otherwise} \end{cases} \quad (3-59)$$

Here,  $p_{bi}$  is the initial structural strength,  $p_{bf}$  the failure structural strength,  $\varepsilon_{p,deve}$  is the equivalent plastic 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,  $\varepsilon_{p,deve}$ , has a typical value between 0.15 and 0.3 for most clays.

Structured clays show a higher void ratio than destructured clays at the same effective mean stress. The virgin compression behavior during the destructuring process is expressed by

$$e = e^* + \Delta e$$

$$\Delta e = \Delta e_i (p_{e0}/p_c)^{d_v}$$

where  $e$  is void ratio of the structured clay,  $\Delta e$  is additional void ratio,  $e^*$  is void ratio of the destructured clay at the same stress state. The void ratio can be found by using Equation 3-57. Further,  $\Delta e_i$  is the additional void ratio at consolidation pressure, and  $d_v$  is the destructuring index due to volumetric deformation.

The yield surface for the modified structured Cam-Clay model reads

$$F_y = q^2 + M^2(p + p_b)(p - p_c)$$

The nonassociated plastic potential reads

$$Q_p = q^2 + \frac{M^2}{\zeta - 1} \left( (p + p_b)^2 - \left( \frac{p + p_b}{p_c + p_b} \right)^{2/\zeta} (p_c + p_b)^2 \right)$$

Here,  $\zeta$  is a parameter to smooth the shape of the plastic potential. The slope of the critical state line,  $M$  can be found by using [Equation 3-52](#).

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 is described as

$$\dot{p}_c = \begin{cases} -\frac{1 + e_0}{\lambda^* - \kappa + d_v \Delta e} p_c \dot{\epsilon}_{p,vol} & \text{if } \eta < M \\ -\frac{1 + e_0}{\lambda^* - \kappa + d_v \Delta e} p_c \dot{\epsilon}_{p,vol} & \text{otherwise} \end{cases}$$

Here,  $\lambda^*$  is the compression index for destructured clay,  $\kappa$  is the swelling index, and  $\eta$  is ratio of shear stress to mean stress.



The MSCC and MCC models are equivalent when  $\Delta e_i = 0$ ,  $\zeta = 2$ , and  $p_b = 0$ .

The relationship between hydrostatic pressure and volumetric elastic strain is the same as the one outlined in [The Modified Cam-Clay Soil Model](#),

$$p = p_i e^{-B_{el} \epsilon_{el,vol}} \quad \text{with } B_{el} = \frac{1 + e_0}{\kappa} \quad (3-60)$$

The stress tensor is then computed from

$$\sigma = \text{dev}(\sigma_0) + p_i e^{-B_{el} \epsilon_{el,vol}} + 2G \text{dev}(\epsilon_{cl})$$

At zero volumetric strain, and with zero contribution from **Initial Stress and Strain** or **External Stress** features, the pressure in the MSCC model is equal to the reference pressure (see [Equation 3-60](#) and [Equation 3-55](#)). This is needed as the MSCC model do 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. 24). 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 of the original BBM model (Ref. 25). The model implemented in COMSOL Multiphysics follows Ref. 25 with certain modifications described in this section.

The BBMx model implemented in the Geomechanics Module presents a smooth yield surface with respect to both stress and suction

$$F_y = q^2 + M^2(p - p_{cs})(p + p_s) + p_{ref}^2 \left( e^{\frac{b(s-s_v)}{p_{ref}}} - e^{\frac{-bs_v}{p_{ref}}} \right)$$

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 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 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,  $\lambda_0$  is the compression index at saturation, and  $\kappa$  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.

As opposed to [The Modified Cam-Clay Soil Model](#), the slope of the critical state line,  $M$ , depends on both the Lode angle  $\theta$  and the angle of internal friction  $\phi$ .

$$M = \left( \frac{6 \sin(\phi)}{3 - \sin(\phi)} \right) \left( \frac{2\omega}{1 + \omega - \sin\left(3\left(\theta - \frac{\pi}{6}\right)\right)(1 - \omega)} \right)^{1/4}$$

where

$$\omega = \left( \frac{3 - \sin(\phi)}{3 + \sin(\phi)} \right)^4$$

The associated plastic potential for the BBMx model reads

$$Q_p = F_y$$

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  $\epsilon_{p,\text{vol}}$ .

$$\dot{p}_c = -\frac{1 + e_0}{\lambda_0 - \kappa} p_c \dot{\epsilon}_{p,\text{vol}} \quad (3-61)$$

The initial void ratio  $e_0$ , the swelling index  $\kappa$ , and the compression index at saturation  $\lambda_0$ , are positive parameters.

The evolution of the yield value at current suction,  $s_y$ , is also governed by the volumetric plastic strain  $\epsilon_{p,\text{vol}}$  as

$$\dot{s}_y = -\frac{1 + e_0}{\lambda_s - \kappa_s} (s_y + p_{\text{atm}}) \dot{\epsilon}_{p,\text{vol}} \quad (3-62)$$

Here,  $\lambda_s$  is the compression index for changes in suction,  $\kappa_s$  is the swelling index for changes in suction, and  $p_{\text{atm}}$  is the atmospheric pressure, all positive parameters. Note that [Equation 3-61](#) and [Equation 3-62](#) are normally given with opposite sign, but here we use the structural mechanics convention so the increments in consolidation pressure and suction are positive in compression.

The evolution of void ratio is then written as

$$e = (1 + e_0)\varepsilon_{\text{vol}} + e_0 \text{ or}$$

$$e = e_{s, \text{ref}} - \lambda(s) \ln\left(\frac{P_{cs}}{P_{\text{ref}}}\right) + \kappa \ln\left(\frac{P_{cs}}{p_1}\right) - \kappa \ln\left(\frac{p}{p_1}\right)$$

where

$$e_{s, \text{ref}} = e_{\text{ref}} - \kappa_s \ln((s + p_{\text{atm}})/p_{\text{atm}})$$

In the BBMx model, the total volumetric elastic response is combination of elastic response by pressure and suction,

$$\varepsilon_{\text{el,vol}} = \varepsilon_{\text{el,vol}}^{\sigma} + \varepsilon_{\text{el,vol}}^s = -\frac{dp}{K} - \frac{ds}{K_{cs}}$$

where

$$K_{cs} = (1 + e_0)(s + p_{\text{atm}})/\kappa_s$$

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

$$\varepsilon_{\text{el,vol}}^s = -\frac{\kappa_s}{1 + e_0} \ln\left(\frac{s + p_{\text{atm}}}{s_0 + p_{\text{atm}}}\right)$$

Here,  $s_0$  is the initial suction. The volumetric elastic response due to suction is accounted in the total strain tensor for BBMx model, so any changes in suction with keeping pressure constant can cause volumetric strains.

The relationship between hydrostatic pressure and volumetric elastic strain is the same as the one outlined in [The Modified Cam-Clay Soil Model](#)

$$p = p_1 e^{-B_{\text{el}} \varepsilon_{\text{el,vol}}} \text{ with } B_{\text{el}} = \frac{1 + e_0}{\kappa} \quad (3-63)$$

The stress tensor is then computed from

$$\sigma = \text{dev}(\sigma_0) + p_1 e^{-B_{\text{el}} \varepsilon_{\text{el,vol}}} + 2G \text{dev}(\varepsilon_{\text{cl}})$$

At zero volumetric strain, and with zero contribution from **Initial Stress and Strain** or **External Stress** features, the pressure in the BBMx model is equal to the reference pressure (see Equation 3-63 and Equation 3-55). This is needed as 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).



Note that the material property  $\lambda_0$  is the compression index at saturation, which does not depend on the suction. The variable  $\lambda(s)$ , which is a function of the current suction, is the compression index (slope) in the void ratio vs. logarithm of the mean stress plot. The material property  $\lambda_s$  is the compression index (slope) in the void ratio vs. logarithm of the matrix suction plot, 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 its stiffness is different during the primary loading and unloading/reloading cases (Ref. 26). The yield surface for the Hardening Soil model is a combination of a conical surface and an elliptic cap surface in stress space. Failure in shear occurs according to a Mohr–Coulomb criterion.

The stiffness moduli for primary loading, denoted by  $E_{50}$ , and for unloading/reloading, denoted by  $E_{ur}$ , 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

$$E_{ur} = E_{ur}^{\text{ref}} \left( \frac{c \cot(\phi) - \sigma_1}{c \cot(\phi) + p_{\text{ref}}} \right)^m$$

Here  $E_{50}^{\text{ref}}$  and  $E_{ur}^{\text{ref}}$  are reference stiffness moduli at reference pressure  $p_{\text{ref}}$ ;  $c$  is the cohesion,  $\phi$  is the angle of internal friction, 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, and defined as

$$q_f = \frac{2c \cos(\phi) - 2\sigma_1 \sin(\phi)}{1 - \sin(\phi)} \quad \text{and} \quad q_a = \frac{q_f}{R_f}$$

where  $R_f$  is the failure ratio. Consider the stress invariants  $q = \sqrt{3J_2}$  and  $p = -I_1/3$ , the yield function and plastic potential for the shear hardening cone are given by

$$F_y = q_a \left( \frac{q_a q}{E_{50}(q_a - q)} - \frac{2q}{E_{ur}} - \gamma_p \right)$$

$$Q_p = \frac{q}{2} \left( p + \frac{q}{6} \right) \sin(\psi_m)$$

where  $\gamma_p$  is the accumulated plastic shear strain and  $\psi_m$  is the mobilized dilatancy angle.



The definition of the shear strain measure  $\gamma_p$  in the original *Hardening Soil* model is not compatible with pure volumetric loading as it does not vanish during pure volumetric straining. Therefore, the shear strain measure  $\gamma_p$  is defined as  $\gamma_p = 2\sqrt{J_2(\epsilon_{pl})}$ , see [Nonlinear Elastic Materials](#).

The yield function for the elliptic cap, and the associated plastic potential, are also defined in terms of stress invariants, and given by

$$F_y = Q_p = \frac{(R_c \bar{q})^2}{p_c} + \frac{p^2}{p_c} - p_c$$

Here,  $R_c$  is the ellipse aspect ratio, and is given by

$$R_c = \frac{1}{K^{nc}_0}$$

where

$$K^{nc}_0 = 1 - \sin(\phi)$$

The special deviatoric stress  $q$  is defined as

$$\bar{q} = \delta \sigma_1 - (\delta - 1) \sigma_2 - \sigma_3$$

where

$$\delta = \frac{3 + \sin(\phi)}{3 - \sin(\phi)}$$

The internal variables  $p_c$  and  $\gamma_p$  depends on the volumetric plastic strain  $\epsilon_{pl,vol}$  and the plastic strain invariant  $J_2(\epsilon_{pl})$ , and their evolution is defined as

$$\dot{p}_c = -H\dot{\epsilon}_{pl,vol}$$

$$\dot{\gamma}_p = 2\sqrt{J_2(\dot{\epsilon}_{pl})}$$

where  $H$  is the hardening modulus. The hardening modulus depends on the bulk modulus in compression  $K_c$  and the bulk modulus in swelling  $K_s$ , and is derived from

$$\frac{1}{H} = \frac{1}{K_c} - \frac{1}{K_s} \text{ where } K_s = \frac{E_{ur}^{ref}}{3(1-2\nu_{ur})}$$

The dilatancy cutoff is implemented by setting the mobilized dilatancy angle  $\psi_m$  equal to zero when the void ratio reaches the critical void ratio  $e_{max}$ .

For an additive decomposition of strains, Cauchy's stress tensor is written as

$$\boldsymbol{\sigma} - \boldsymbol{\sigma}_0 = \mathbf{C}:(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{inel})$$

where  $\mathbf{C}$  is a function of the stiffness modulus  $E_{ur}$  and Poisson's ratio  $\nu$ .

## *Creep and Viscoplasticity*

---

### **ABOUT CREEP**



In the literature, the terms *viscoplasticity* and *creep* are often used interchangeably to refer to the class of problems related to rate-dependent plasticity.

Creep is an inelastic time-dependent deformation that occurs when a material is subjected to stress (typically much less than the yield stress) at sufficiently high temperatures.

The creep strain rate, in a general case, depends on stress, temperature, and time, usually in a nonlinear manner:

$$\dot{\epsilon}_{cr} = F_{cr}(\boldsymbol{\sigma}, T, t)$$

It is often possible to separate these effects as shown in this equation:

$$F_{cr}(\sigma, t, T) = f_1(\sigma)f_2(T)f_3(t)$$

Experimental data shows three types of behavior for the creep strain rate at constant stress as function of time. Researchers normally subdivide the creep curve into 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

$$\dot{\epsilon}_{cr} = F_{cr1} + F_{cr2} + F_{cr3}$$

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.

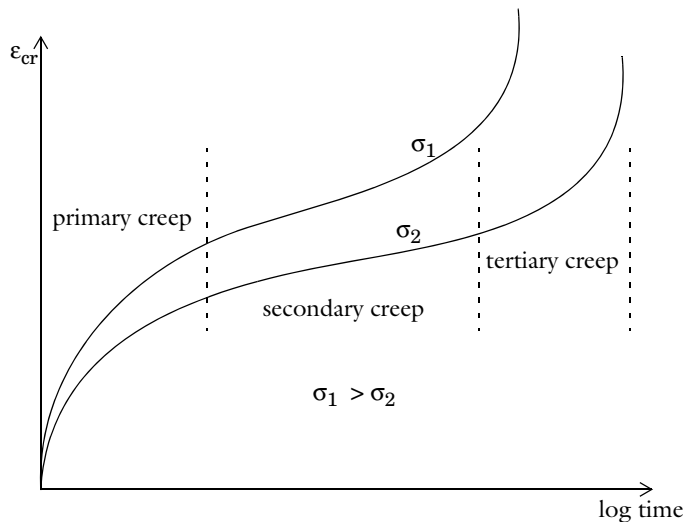


Figure 3-25: Uniaxial creep as a function of logarithmic time.

### *Creep and Viscoplastic Material Models*

Despite the fact that the creep response of a given material is related to its atomic structure, a macroscopic (continuum mechanics) description is normally appropriate for modeling scientific and engineering problems.

In COMSOL Multiphysics, there are several creep models. These models can be split into two main groups. One set of models are more general, and you will have to express the creep rate yourself, based on other variables as a stress tensor and temperature. These models are:

- Creep Potential
- Volumetric Creep
- Deviatoric Creep
- User Defined Creep

In addition to the basic models for creep described above, there are also predefined material models for creep in metals and crystalline solids:

- Norton Law (Power law)
- The Norton–Bailey Law
- Garofalo Law (Hyperbolic Sine Law)
- Nabarro–Herring Creep (Diffusional Creep)
- Coble Creep (Diffusional Creep)
- Weertman Creep (Dislocation Creep)
- Anand Viscoplastic Model
- Chaboche Viscoplastic Model
- Perzyna Viscoplastic Model

All creep models are contributing subnodes to a basic material model like **Linear Elastic Material** and they can be combined with any other subnodes, such as **Plasticity** or **Thermal Expansion** to create more advanced models. They can also be combined with each other to model several creep mechanisms acting at the same time.

### **CREEP POTENTIAL**

Some authors use a creep potential to describe the secondary creep rate, so that the creep rate is written in a way similar to the flow rule for plasticity:

$$\dot{\epsilon}_{cr} = \eta \frac{\partial Q_{cr}}{\partial \sigma} \text{ and } \eta > 0$$



Here,  $Q_{cr}$  is a user defined creep potential, which is normally written in terms of invariants of the stress tensor.

Volumetric creep is obtained when the creep potential depends only on the first invariant of Cauchy stress tensor,  $I_1(\sigma)$ , since

$$\frac{\partial Q_{cr}}{\partial \sigma} = \frac{\partial Q_{cr}}{\partial I_1} \mathbf{I}$$

This is equivalent to that the creep potential would depend on the pressure  $p = -I_1/3$ .

When the creep potential depends only on the second deviatoric invariant of Cauchy stress tensor,  $J_2(\sigma)$ , the deviatoric creep model is obtained since

$$\frac{\partial Q_{cr}}{\partial \sigma} = \frac{\partial Q_{cr}}{\partial J_2} \text{dev}(\sigma)$$

This is equivalent to that the creep potential would depend on the equivalent stress  $\sigma_e = \sqrt{3J_2}$ .

When the creep potential,  $Q_{cr}$ , is given in units of Pa, the rate multiplier  $\eta$  is given in units of 1/s.

### **VOLUMETRIC CREEP**

The creep strain rate is calculated by solving the rate equation

$$\dot{\epsilon}_{cr} = \frac{1}{3} \dot{\epsilon}_{cr,vol} \mathbf{I}$$

so that the creep rate tensor is a diagonal tensor. The trace of the creep strain rate tensor, the *volumetric creep strain rate*, equals the user input  $\dot{\epsilon}_{cr,vol}$

$$\text{trace}(\dot{\epsilon}_{cr}) = \dot{\epsilon}_{cr,vol}$$

The volumetric creep strain rate usually depends on the first invariant of Cauchy stress  $I_1(\sigma)$  or the pressure  $p = -I_1/3$ , in addition to the temperature and other material parameters.

Volumetric creep is not generally used to model creep in metals, but it is commonly used to model creep in soils or other geological materials.

### **DEVIATORIC CREEP**

The creep strain rate is calculated by solving the rate equation

$$\dot{\epsilon}_{cr} = \dot{\epsilon}_{cre} \mathbf{n}^D$$

Here,  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor.

The *equivalent creep strain rate*,  $\dot{\epsilon}_{cre}$ , normally depends on the second deviatoric invariant of the stress  $J_2(\boldsymbol{\sigma})$  or the equivalent von Mises stress  $\sigma_{mises}$ , in addition to the temperature and other material parameters.

The deviatoric tensor  $\mathbf{n}^D$  is defined as

$$\mathbf{n}^D = \frac{3 \operatorname{dev}(\boldsymbol{\sigma})}{2 \sigma_e} \quad (3-64)$$

The resulting creep strain rate tensor is also deviatoric, since  $\operatorname{trace}(\mathbf{n}^D) = 0$

$$\operatorname{trace}(\dot{\epsilon}_{cr}) = \dot{\epsilon}_{cre} \operatorname{trace}(\mathbf{n}^D) = 0$$

Given the property

$$\mathbf{n}^D : \mathbf{n}^D = \frac{3}{2}$$

the *equivalent creep strain rate* equals the absolute value of the user input

$$\dot{\epsilon}_{cre} = \sqrt{\frac{2}{3} \dot{\epsilon}_{cr} : \dot{\epsilon}_{cr}}$$

Deviatoric creep is very popular to model creep in metals and alloys. For example, Norton's law is a deviatoric creep model.



The equivalent creep strain and the equivalent creep strain rate are available in the variables `solid.ece` and `solid.ecet`.

---

### USER DEFINED CREEP

The creep strain tensor is calculated by time-integration the user defined symmetric creep strain rate tensor  $\dot{\epsilon}_{cr}$ .

### NORTON LAW (POWER LAW)

The most common model for secondary creep is the *Norton equation* where the creep strain rate is proportional to a power of the equivalent stress,  $\sigma_e$ :

$$\dot{\epsilon}_{cr} \propto \sigma_e^n$$

This is normally true at intermediate to high stress levels and at absolute temperatures of  $T/T_m > 0.5$ , where  $T_m$  is the melting temperature (that is, the temperature in the solid is at least as high as half the melting temperature  $T_m$ ). An “Arrhenius type” temperature dependency can also be included. It is defined by

$$\dot{\epsilon}_{\text{cr}} \propto e^{-Q/RT}$$

where  $Q$  is the activation energy (SI unit: J/mol),  $R$  is the gas constant, and  $T$  is the absolute temperature (SI unit: K).

Norton creep is a deviatoric temperature-dependent creep model, with a creep rate equation written as

$$\dot{\epsilon}_{\text{cr}} = A \left( \frac{\sigma_c}{\sigma_{\text{ref}}} \right)^n e^{-\frac{Q}{RT}} \mathbf{n}^D \quad (3-65)$$

Here,  $A$  is the creep rate coefficient (SI unit: 1/s),  $n$  is the stress exponent (dimensionless),  $\sigma_{\text{ref}}$  a reference stress level (SI unit: Pa), and  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in [Equation 3-64](#).



- For a discussion about how to convert common creep data equations into the form used in COMSOL Multiphysics, see [Converting Between Different Creep Data Representations](#)
- See also the description of the [Norton](#) material model in the Solid Mechanics interface documentation.

### THE NORTON–BAILEY LAW

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 time and to a power of the equivalent stress

$$\epsilon_{\text{cr}} \propto \sigma_e^n t^m$$

which for the creep strain rate becomes a time hardening formulation of Norton’s law. Differentiating with respect to time will give the rate form.

$$\dot{\epsilon}_{\text{cr}} \propto \sigma_e^n m t^{m-1}$$

Norton–Bailey creep is a deviatoric temperature-dependent creep model, furnished with either a time-hardening or a strain-hardening primary creep model. The creep rate equation for the time-hardening model used in COMSOL Multiphysics is written as

$$\dot{\boldsymbol{\varepsilon}}_{\text{cr}} = F_{\text{cr}} m \left( \frac{t + t_{\text{shift}}}{t_{\text{ref}}} \right)^{m-1} \mathbf{n}^D \quad (3-66)$$

where  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-64, and  $F_{\text{cr}}$  is expressed as in the Norton model:

$$F_{\text{cr}} = A \left( \frac{\sigma_c}{\sigma_{\text{ref}}} \right)^n e^{-\frac{Q}{RT}} \quad (3-67)$$

Here,  $A$  is the creep rate coefficient (SI unit: 1/s),  $n$  is the stress exponent (dimensionless),  $\sigma_{\text{ref}}$  is a reference stress level (SI unit: Pa),  $t_{\text{ref}}$  and  $t_{\text{shift}}$  are the reference and shift times (SI unit: s), and  $m$  is the time-hardening exponent (dimensionless).

The strain-hardening variant of this creep law is implemented as

$$\dot{\boldsymbol{\varepsilon}}_{\text{cr}} = F_{\text{cr}} m \left( \frac{\varepsilon_{\text{cre}} + \varepsilon_{\text{shift}}}{t_{\text{ref}} F_{\text{cr}}} \right)^{\frac{m-1}{m}} \mathbf{n}^D \quad (3-68)$$

where  $\varepsilon_{\text{cre}}$  is the equivalent creep strain, and  $\varepsilon_{\text{shift}}$  is the equivalent creep strain shift.

The time and frequency shifts in Equation 3-66 and Equation 3-68 serve two purposes:

- They can be used to initialize a study where some hardening has already taken place.
- 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 problems for the numerical solution. You can then add a small shift to overcome this problem.



- For a discussion about how to convert common creep data equations into the form used in COMSOL Multiphysics, see [Converting Between Different Creep Data Representations](#)
- See also the description of the [Norton–Bailey](#) material model in the Solid Mechanics interface documentation.

### **GAROFALO LAW (HYPERBOLIC SINE LAW)**

At very high stress levels, the creep rate is proportional to the exponential of the equivalent stress

$$\dot{\epsilon}_{\text{cr}} \propto e^{\alpha\sigma_e}$$

Garofalo showed ([Ref. 8](#), [Ref. 9](#)) that the power-law and exponential creep are limiting cases for the general empirical expression

$$\dot{\epsilon}_{\text{cr}} \propto (\sinh(\alpha\sigma_e))^n$$

This equation reduces to a power law (the Norton law) for  $\alpha\sigma_e < 0.8$  and approaches exponential creep for  $\alpha\sigma_e > 1.2$ , where  $1/\alpha$  is a reference equivalent stress level.

Garofalo creep is also a deviatoric creep model with a creep rate proportional to the hyperbolic sine function. It can also be augmented by an “Arrhenius type” temperature dependency such that

$$\dot{\epsilon}_{\text{cr}} \propto e^{-Q/RT}$$

where  $Q$  is the activation energy (SI unit: J/mol),  $R$  is the gas constant, and  $T$  is the absolute temperature (SI unit: K). The complete creep rate equation as used in COMSOL Multiphysics then reads

$$\dot{\epsilon}_{\text{cr}} = A \left( \sinh \left( \frac{\sigma_e}{\sigma_{\text{ref}}} \right) \right)^n e^{-\frac{Q}{RT}} \mathbf{n}^D$$

where,  $A$  is the creep rate (SI unit: 1/s),  $n$  is the stress exponent (dimensionless), and  $\sigma_{\text{ref}}$  a reference equivalent stress level (SI unit: Pa).  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in [Equation 3-64](#).



- For a discussion about how to convert common creep data equations into the form used in COMSOL Multiphysics, see [Converting Between Different Creep Data Representations](#)
- See also the description of the [Garofalo \(Hyperbolic Sine\)](#) material model in the Solid Mechanics interface documentation.

### NABARRO–HERRING CREEP (DIFFUSIONAL CREEP)

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

$$\dot{\epsilon}_{\text{cr}} = \frac{7D_v b^3}{k_B T d^2} \sigma_c \mathbf{n}^D$$

Here,  $d$  is the grain diameter,  $D_v$  is the volume diffusivity through the grain interior,  $b$  is Burgers vector,  $k_B$  is the Boltzmann's constant, and  $T$  is the absolute temperature.  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in [Equation 3-64](#).



See also the description of the [Nabarro–Herring](#) material model in the Solid Mechanics interface documentation.

### COBLE CREEP (DIFFUSIONAL CREEP)

Coble creep ([Ref. 6](#), [Ref. 7](#)) is closely related to Nabarro–Herring creep but takes into account the grain boundary diffusivity,  $D_{\text{gb}}$

$$\dot{\epsilon}_{\text{cr}} = \frac{50D_{\text{gb}} b^4}{k_B T d^3} \sigma_c \mathbf{n}^D$$



Coble creep is more sensitive to grain diameter than Nabarro–Herring creep.



See also the description of the [Coble](#) material model in the physics interface documentation.

### WEERTMAN CREEP (DISLOCATION CREEP)

At intermediate to high stress levels and temperatures  $T/T_m > 0.5$ , the creep mechanism is assumed to be diffusion-controlled movements of dislocations in the crystal lattices ([Ref. 7](#))

$$\dot{\epsilon}_{\text{cr}} = \frac{Db}{k_B T} \sigma_{\text{ref}} \left( \frac{\sigma_e}{\sigma_{\text{ref}}} \right)^n \mathbf{n}^D$$

where  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in [Equation 3-64](#). Generally, the stress exponent  $n$  takes values between 3 and 5.

A general relation between creep rate and several material parameters is the *Mukherjee–Bird–Dorn* equation ([Ref. 6](#))

$$\dot{\epsilon}_{\text{cr}} \propto \frac{DGb}{k_B T} \left( \frac{\sigma_e}{G} \right)^n \left( \frac{b}{d} \right)^p e^{-Q/RT}$$

Here,  $T$  is the temperature,  $d$  is the grain size,  $b$  is the Burgers vector,  $D$  is the self-diffusion coefficient,  $G$  is the shear modulus, and  $e^{-Q/RT}$  is an “Arrhenius” type of temperature dependency.

For high temperatures, *Mukherjee–Bird–Dorn* equation describes Weertman creep when setting  $p = 0$ . Setting  $n = 0$  and  $p = 2$  describes Nabarro–Herring, and setting  $n = 0$  and  $p = 3$  describes Coble creep. Harper–Dorn creep is obtained by setting  $n = 1$  and  $p = 0$ .



See also the description of the [Weertman](#) material model in the Solid Mechanics interface documentation.

### ANAND VISCOPLASTIC MODEL

The Anand viscoplasticity ([Ref. 9](#)) is a deviatoric creep model suitable for large, isotropic, viscoplastic deformations in combination with small elastic deformations.

The viscoplastic strain rate equation reads

$$\dot{\epsilon}_{\text{cr}} = F_{\text{cr}} \mathbf{n}^D$$

where  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-64, and the creep rate is calculated from

$$F_{\text{cr}} = A e^{-Q/RT} \left[ \sinh \left( \xi \frac{\sigma_c}{s_a} \right) \right]^{\frac{1}{m}}$$

Here,  $A$  is the creep rate coefficient (SI unit:  $s^{-1}$ ),  $Q$  is the activation energy (SI unit: J/mol),  $m$  is the stress sensitivity,  $\xi$  is the multiplier of stress,  $R$  is the gas constant, and  $T$  is the absolute temperature (SI unit: K).

The internal variable,  $s_a$ , is called *deformation resistance* (SI unit: Pa) and is calculated from the rate equation

$$\dot{s}_a = h_0 \left| 1 - \frac{s_a}{s_a^*} \right|^{a-1} \left( 1 - \frac{s_a}{s_a^*} \right) F_{\text{cr}}$$

with the initial condition  $s_a(0) = s_{\text{init}}$ . Here,  $h_0$  is the *hardening constant* (SI unit: Pa), and  $a$  is the *hardening sensitivity*.

The variable  $s_a^*$  is the saturation value of the deformation resistance  $s_a$ , which is calculated from the expression

$$s_a^* = s_0 \left( \frac{F_{\text{cr}}}{A} e^{-Q/RT} \right)^n$$

where  $s_0$  is the *deformation resistance saturation coefficient* (SI unit: Pa), and  $n$  is the *deformation resistance sensitivity*.

### CHABOCHE VISCOPLASTIC MODEL

The viscoplastic strain rate tensor is given by

$$\dot{\epsilon}_{\text{vp}} = A \left\langle \frac{F_y}{\sigma_{\text{ref}}} \right\rangle^n \mathbf{n}^D$$

Here,  $A$  is the viscoplastic rate coefficient (SI unit: 1/s),  $n$  is the stress exponent (dimensionless),  $\sigma_{\text{ref}}$  a reference stress level (SI unit: Pa), and  $\mathbf{n}^D$  is a deviatoric tensor coaxial to the stress tensor. The Macaulay brackets are applied on the yield function, which is defined as done for plasticity

$$F_y = \phi(\sigma) - \sigma_{ys}$$



The equivalent stress  $\phi(\sigma)$  is either the von Mises, Tresca, or Hill stress; or a user defined expression, and  $\sigma_{ys}$  is the yield stress (which may include a linear or nonlinear [Isotropic Hardening](#) model). The stress tensor used in the equivalent stress  $\phi(\sigma)$  is shifted by what is usually called the *back stress*,  $\sigma_{back}$  when [Kinematic Hardening](#) is included.

The deviatoric tensor  $\mathbf{n}^D$  is computed from the plastic potential  $Q_p$

$$\mathbf{n}^D = \frac{\partial Q_p}{\partial \sigma}$$

When von Mises equivalent stress is used, the associated flow rule reads  $Q_p = F_y$ , and the deviatoric tensor  $\mathbf{n}^D$  is defined as done for deviatoric creep

$$\mathbf{n}^D = \frac{3 \operatorname{dev}(\sigma)}{2 \sigma_{mises}} \quad (3-69)$$

Given the property

$$\mathbf{n}^D : \mathbf{n}^D = \frac{3}{2}$$

the equivalent viscoplastic strain rate is equivalent to

$$\dot{\epsilon}_{vpc} = \sqrt{\frac{2}{3}} \dot{\epsilon}_{vp} : \dot{\epsilon}_{vp} = A \left\langle \frac{F_y}{\sigma_{ref}} \right\rangle^n$$

#### PERZYNA VISCOPLASTIC MODEL

Perzyna viscoplastic model is similar to the Chaboche model, with the exception than the stress exponent is set equal to one. The viscoplastic strain rate tensor is then given by

$$\dot{\epsilon}_{vp} = \frac{A}{\sigma_{ref}} \left\langle F_y \right\rangle \mathbf{n}^D$$

Some authors denote the viscosity as the quotient  $\eta = \sigma_{ref}/A$ .



See also the description of [Viscoplasticity](#) in the physics interface documentation.

---

## ENERGY DISSIPATION

Since creep and viscoplasticity are inelastic processes, the dissipated energy density can be calculated by integrating the *creep dissipation rate density* (SI unit:  $\text{W}/\text{m}^3$ ) given by

$$\dot{W}_{\text{cdr}} = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}_{\text{cr}}$$

In case many creep sub-nodes are added, the creep dissipation rate density is calculated from the cumulative creep strain rate tensor  $\dot{\boldsymbol{\epsilon}}_{\text{cr}}$ .

The total energy dissipated in a given volume can be calculated by a volume integration of the dissipated creep energy density  $W_{\text{e}}$  (SI unit:  $\text{J}/\text{m}^3$ ).



When the **Calculate dissipated energy** check box 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.Wvldr`. 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.

## CONVERTING BETWEEN DIFFERENT CREEP DATA REPRESENTATIONS

The equation forms described for the different creep models above differ from the forms most commonly found in the literature. The difference lies in the introduction of normalizing reference values such as the reference stress  $\sigma_{\text{ref}}$  and reference time  $t_{\text{ref}}$ . 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. This system has two advantages:

- It is possible to use the built-in unit management system in COMSOL Multiphysics.
- You do not need to do any difficult unit conversions when creep data are available in units other than the model units. 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  $A_{\text{N}}$  and  $n$  in the equation

$$\dot{\boldsymbol{\epsilon}}_{\text{cr}} = A_{\text{N}} \boldsymbol{\sigma}_{\text{e}}^n$$

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 form used in COMSOL Multiphysics (Equation 3-65) requires the introduction of the reference stress  $\sigma_{\text{ref}}$ . It is here convenient to use the implicit stress unit for which  $A_N$  is given as 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,  $\sigma_{c7}$  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 to the value of  $\sigma_{c7}$  and enter the creep rate coefficient as  $1\text{e-}7[1/\text{h}]$ .

### Example

Assume that a carbon steel has the following two equivalent descriptions of its creep properties at a certain temperature:

- $\sigma_{c7} = 70$  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:

- $\sigma_{\text{ref}}$  as 70 [MPa]
- $n$  as 4.5
- $A$  as  $1\text{e-}7[1/\text{h}]$ .

In the second case, enter:

- $\sigma_{\text{ref}}$  as 1 [MPa]
- $n$  as 4.5
- $A$  as  $4.98\text{e-}16[1/\text{h}]$

These two sets of data describe the same material.

### *The Norton–Bailey Law*

Material data for a Norton–Bailey law usually is usually written in terms of the creep strain, rather than the creep strain rate, so that the form of the constitutive relation is

$$\epsilon_{\text{cr}} = A_{\text{NB}} \sigma_e^{n_t m}$$

In this case, the coefficient  $A_{\text{NB}}$  has implicit dimension and units which depend on the values of  $n$  and  $m$ , and on the stress and time units.

Converting the data to the form used in COMSOL Multiphysics (Equation 3-66 and Equation 3-67) requires the introduction of both an arbitrary reference stress  $\sigma_{\text{ref}}$  and an arbitrary reference time  $t_{\text{ref}}$ . If you use the implicit units for which  $A_{\text{NB}}$  is given as the reference values, then the constant  $A$  will have the same numerical value as  $A_{\text{NB}}$ .

#### *Garofalo Law*

Since the stress inside in the Garofalo law appears as an argument to a  $\sinh()$  function, it must necessarily be nondimensionalized. Most commonly this is however written as

$$\dot{\epsilon}_{\text{cr}} \propto (\sinh(\alpha \sigma_e))^n$$

Comparing with the expression in COMSOL Multiphysics,

$$\dot{\epsilon}_{\text{cr}} \propto \left( \sinh\left(\frac{\sigma_e}{\sigma_{\text{ref}}}\right) \right)^n$$

it is evident that the reference stress should be chosen as

$$\sigma_{\text{ref}} = \frac{1}{\alpha}$$

In this case, there is no arbitrariness in the choice of  $\sigma_{\text{ref}}$ , since  $\alpha$  is an actual material parameter.

#### *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.

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*.

Cold NiTi alloys are composed by 100% of martensite volume fraction,  $\xi_{\text{M}}$ . The austenite phase develops upon heating the material above the *austenite start temperature*,  $A_s$ . If the alloy is heated above the *austenite finish temperature*,  $A_f$ , the

alloy microstructure becomes 100% of austenite. The reverse process has different transition temperatures: If a 100% austenite alloy is cooled below the *martensite start temperature*,  $M_s$ , the martensite volume fraction  $\xi_M$  will develop until the cooling temperature reaches the *martensite finish temperature*,  $M_f$ , below which the alloy microstructure will be pure martensite.

Many industrial applications take advantage of this hysteresis loop, as the transition temperatures are not the same in a heating-cooling cycle.

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  $\boldsymbol{\varepsilon}$  and the temperature  $T$ . An additional internal variable is used to compute the *transformation strain tensor*  $\boldsymbol{\varepsilon}_{tr}$  (Ref. 23-24)

$$\Psi(\boldsymbol{\varepsilon}, T, \boldsymbol{\varepsilon}_{tr}) = \rho c T (1 - \ln(T)) + \frac{K}{2} \boldsymbol{\varepsilon}_{vol}^2 + G (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{tr})^2 + \frac{H_k}{2} (\boldsymbol{\varepsilon}_{tr})^2 + f(T) |\boldsymbol{\varepsilon}_{tr}| + I(\boldsymbol{\varepsilon}_{tr})$$

here,  $c$  is the heat capacity at constant pressure,  $K$  and  $G$  are the bulk and shear moduli,  $\boldsymbol{\varepsilon}_{vol}$  is the volumetric strain,  $H_k$  is the hardening modulus, and  $I(\boldsymbol{\varepsilon}_{tr})$  is the *indicator function* for the strain limit constraint. The *equivalent transformation strain*  $\boldsymbol{\varepsilon}_{tre}$  is used as a measure of the transformation strain tensor

$$\boldsymbol{\varepsilon}_{tre} = |\boldsymbol{\varepsilon}_{tr}| = \sqrt{\frac{2}{3} \boldsymbol{\varepsilon}_{tr} : \boldsymbol{\varepsilon}_{tr}}$$

The indicator function is defined by

$$I(\boldsymbol{\varepsilon}_{tr}) = \begin{cases} 0 & \text{for } |\boldsymbol{\varepsilon}_{tr}| < \boldsymbol{\varepsilon}_{tr, \max} \\ \infty & \text{otherwise} \end{cases}$$

The term  $\Psi_{ch} = f(T) |\boldsymbol{\varepsilon}_{tr}|$  is denoted as the chemical energy density due to the thermally induced martensite transformation. The function  $f(T)$  corresponds to the temperature-dependent martensite to austenite equilibrium stress, defined from the slope of the limit curve  $\beta$  and the reference temperature  $T^*$

$$f(T) = \beta \langle T - T^* \rangle$$

Here, the operator  $\langle \rangle$  denotes the Macaulay brackets.

The conjugated thermodynamic stress  $\sigma_{\text{tr}}$  associated to the transformation strain variable is

$$\sigma_{\text{tr}} = \frac{\partial \Psi}{\partial \epsilon_{\text{tr}}} = \text{dev}(\sigma) - H \epsilon_{\text{tr}} - f(T) \frac{\partial |\epsilon_{\text{tr}}|}{\partial \epsilon_{\text{tr}}} - \frac{\partial I}{\partial \epsilon_{\text{tr}}}$$

The evolution of the transformation strain  $\epsilon_{\text{tr}}$  is given by the so-called *limit function*, which takes the same form as the yield function for metal plasticity.

$$F = |\sigma_{\text{tr}}| - \sigma_{\text{ys}0}$$

The evolution equation for the transformation strain  $\epsilon_{\text{tr}}$  is computed from the flow rule

$$\dot{\epsilon}_{\text{tr}} = \lambda_{\text{p}} \frac{\partial F}{\partial \sigma_{\text{tr}}}$$

where the plastic multiplier  $\lambda_{\text{p}}$  is solved with the Kuhn–Tucker conditions, as done for plasticity, see [Plastic Flow for Small Strains](#).

### LAGOUDAS MODEL

For Lagoudas model, Gibbs free energy density depends on two state variables: the total stress tensor  $\sigma$  and the temperature field  $T$ . Additional internal variables are used to compute the transformation strain tensor  $\epsilon_{\text{tr}}$  and the martensite volume fraction  $\xi$  ([Ref. 25](#))

$$G(\sigma, T, \epsilon_{\text{tr}}, \xi) = c \left( (T - T_0) - T \ln \left( \frac{T}{T_0} \right) \right) - s_0 T - \frac{1}{\rho} \sigma : S : \sigma - \frac{1}{\rho} \sigma : (\epsilon_{\text{th}} - \epsilon_{\text{tr}}) + u_0 + \frac{1}{\rho} f(\xi)$$

Here,  $c$  is the heat capacity at constant pressure,  $s_0$  is the specific entropy at reference state,  $S$  is the compliance matrix,  $\epsilon_{\text{th}}$  is the thermal strain tensor,  $\epsilon_{\text{tr}}$  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–Aurichio model, the evolution equation for the transformation strain  $\epsilon_{\text{tr}}$  is computed from the flow rule

$$\dot{\epsilon}_{\text{tr}} = \xi \Lambda$$

where the normalized *transformation tensor*  $\Lambda$  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. 26. 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,  $\xi$ , is calculated from the previous state to determine the expected increment or decrement of the transformation strain tensor  $\epsilon_{tr}$ . This calculation is computational expensive and it can lead to convergence issues. Since in many applications the transformation direction  $\xi$  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

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b_M \xi^2 + (\mu_1 + \mu_2)\xi & \text{for } \xi > 0 \\ \frac{1}{2}\rho b_A \xi^2 + (\mu_1 - \mu_2)\xi & \text{for } \xi < 0 \end{cases}$$

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  $\mu_1$  and  $\mu_2$  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  $n_4$

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b_M \left( \xi + \frac{\xi^{n_1+1}}{n_1+1} + \frac{(1-\xi)^{n_2+1}}{n_2+1} \right) & \text{for } \xi > 0 \\ \frac{1}{2}\rho b_A \left( \xi + \frac{\xi^{n_3+1}}{n_3+1} + \frac{(1-\xi)^{n_4+1}}{n_4+1} \right) & \text{for } \xi < 0 \end{cases}$$

### EQUIVALENT MATERIAL PROPERTIES

In Lagoudas model, the phase transformation is described by six parameters: the transition temperatures  $M_s, M_f, A_s, A_f$ , 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 constant temperature, it is handy to enter the transition stress levels instead of the transition temperatures.

Let us consider a NiTi alloys at constant temperature composed by 100% of austenite volume fraction. Upon loading, the slope in the stress-strain curve would be the Young's modulus of austenite. The martensite phase starts to develop when the axial stress reaches the *martensite start stress*,  $\sigma_{M_s}$ . If the alloy is loaded further, the slope reduces, this region is called *loading plateau*. Above the *martensite finish stress*,  $\sigma_{M_f}$ , the microstructure becomes 100% of martensite. This process is commonly called the *forward transformation*.

The reverse process has different transition stresses: If the 100% martensite alloy is unloaded, the slope equals the martensite Young's modulus. When the axial stress reaches the *austenite start stress*,  $\sigma_{A_s}$ , the austenite volume fraction  $\xi_A$  starts to develop until the axial stress falls below the *austenite finish stress*,  $\sigma_{A_f}$ , at which level the alloy microstructure is 100% austenite. This process is commonly called the *reverse transformation*. [Figure 3-26](#) illustrates this process.



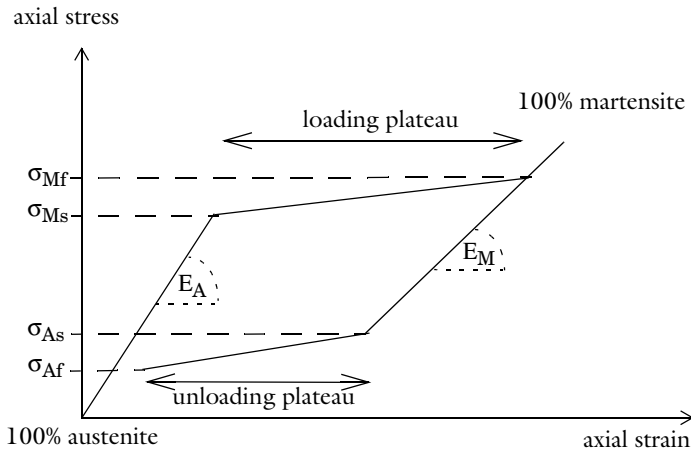


Figure 3-26: Uniaxial stress-strain curve at constant temperature.

The relations between the start and finish temperatures and stresses are

$$\sigma_{Ms} = C_M(T_\sigma - M_s), \quad \sigma_{Mf} = C_M(T_\sigma - M_f)$$

$$\sigma_{As} = C_A(T_\sigma - A_s), \quad \sigma_{Af} = C_A(T_\sigma - A_f)$$

here,  $T_\sigma$  is the constant temperature at which the stress-strain curve was measured.

### *Piezoelectric Material*

---

#### **THE PIEZOELECTRIC EFFECT**

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*

$$\begin{aligned} \mathbf{T} &= c_E \mathbf{S} - e^T \mathbf{E} \\ \mathbf{D} &= e \mathbf{S} + \epsilon_S \mathbf{E} \end{aligned}$$

*Strain-Charge*

$$\begin{aligned} \mathbf{S} &= s_E \mathbf{T} + d^T \mathbf{E} \\ \mathbf{D} &= d \mathbf{T} + \epsilon_T \mathbf{E} \end{aligned}$$

In the above relations, the naming convention used in piezoelectricity theory is assumed, so that the structural strain is denoted by  $\mathbf{S}$ , and the stress is denoted by  $\mathbf{T}$ . 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  $\epsilon$  (instead of  $\mathbf{S}$ ) and the stresses are denoted by either  $\sigma$  or  $\mathbf{S}$  (instead of  $\mathbf{T}$ ). 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

$$\begin{aligned} \sigma &= c_E \epsilon - e^T \mathbf{E} \\ \mathbf{D} &= e \epsilon + \epsilon_0 \epsilon_{rS} \mathbf{E} \end{aligned} \tag{3-70}$$

The stress-charge form is always used in the variational formulation (weak equation form) which COMSOL Multiphysics uses for discretization and computation.

## STRAIN-CHARGE

$$\begin{aligned}\boldsymbol{\varepsilon} &= s_E \boldsymbol{\sigma} + d^T \mathbf{E} \\ \mathbf{D} &= d \boldsymbol{\sigma} + \varepsilon_0 \varepsilon_{rT} \mathbf{E}\end{aligned}\tag{3-71}$$

Most material data appear in the strain-charge form, and it can be easily 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:

$$\begin{aligned}c_E &= s_E^{-1} \\ e &= d s_E^{-1} \\ \varepsilon_S &= \varepsilon_0 \varepsilon_{rS} = \varepsilon_0 \varepsilon_{rT} - d s_E^{-1} d^T\end{aligned}\tag{3-72}$$

You find all the necessary material data inputs within the [Piezoelectric Material](#) feature under the Solid Mechanics interface, which are added automatically when you add 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 3-124](#) with the charge conservation equation of Electrostatics,

$$\nabla \cdot \mathbf{D} = \rho_V\tag{3-73}$$

where the  $\rho_V$  is the electric charge concentration. The electric field is computed from the electric potential  $V$  as

$$\mathbf{E} = -\nabla V$$

In both [Equation 3-73](#) and [Equation 3-124](#), the constitutive relations [Equation 3-70](#) 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:

$$\begin{aligned}\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} &= \nabla \cdot (\mathbf{C} : \nabla \mathbf{u} + \nabla V \cdot \mathbf{e}) \\ \nabla \cdot (\mathbf{e} : \nabla \mathbf{u} - \epsilon_0 \epsilon_r \cdot \nabla V) &= 0\end{aligned}\quad (3-74)$$

The following symmetries hold for the constitutive tensor coefficients:

$$\mathbf{C}^{ijkl} = \mathbf{C}^{jikl} = \mathbf{C}^{klij}, \quad e^{ikl} = e^{ilk}, \quad \epsilon_r^{kl} = \epsilon_r^{lk}.$$



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 electrical enthalpy:

$$H_e(\nabla \mathbf{u}, \nabla V) = \frac{1}{2} \nabla \mathbf{u} : \mathbf{C} : \nabla \mathbf{u} + \nabla V \cdot \mathbf{e} : \nabla \mathbf{u} - \frac{1}{2} \epsilon_0 \nabla V \cdot \epsilon_r \cdot \nabla V$$

Thus,

$$\begin{aligned}\boldsymbol{\sigma} &= \frac{\partial H_e}{\partial \nabla \mathbf{u}} = \mathbf{C} : \nabla \mathbf{u} + \nabla V \cdot \mathbf{e} \\ \mathbf{D} &= \frac{\partial H_e}{\partial \nabla V} = \mathbf{e} : \nabla \mathbf{u} - \epsilon_0 \epsilon_r \cdot \nabla V\end{aligned}$$

Since the relations in [Equation 3-74](#) 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 3-18) 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

$$\Gamma_n = \mathbf{n} \cdot \mathbf{C} \cdot \mathbf{n} + \mathbf{e}_n \otimes \mathbf{e}_n / \varepsilon_n$$

and

$$\mathbf{e}_n = \mathbf{n} \cdot \mathbf{e} \cdot \mathbf{n}$$

$$\varepsilon_n = \varepsilon_0 \mathbf{n} \cdot \boldsymbol{\varepsilon}_r \cdot \mathbf{n}$$

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 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.

However, most piezoelectric materials exhibit only moderate degree of anisotropy. The material data is usually given in a coordinate system, in which it is either orthotropic or even transverse isotropic where 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  $X$ -axis direction is a particular solution, for which

$$\begin{aligned} u &= \hat{u} \exp[-ik_X(X - c_X t)] \\ v &= 0 \\ w &= 0 \\ V &= \hat{V} \exp[-ik_X(X - c_X t)] \end{aligned}$$

The corresponding phase velocity is given by

$$c_{pX} = \frac{1}{\rho} \left( c_{E, 11} + \frac{e_{X1}^2}{\epsilon_0 \epsilon_{rS, XX}} \right)^{1/2}$$

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  $X$ -axis direction and with  $XY$ -plane polarization is a solution such that

$$\begin{aligned} u &= 0 \\ v &= \hat{v} \exp[-ik_X(X - c_X t)] \\ w &= 0 \\ V &= \hat{V} \exp[-ik_X(X - c_X t)] \end{aligned}$$

and the corresponding phase velocity is computed as

$$c_{sXY} = \frac{1}{\rho} \left( c_{E, 66} + \frac{e_{X6}^2}{\epsilon_0 \epsilon_{rS, XX}} \right)^{1/2}$$

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:

$$\frac{\partial \omega_j}{\partial \mathbf{k}} = \frac{1}{2\rho c_j} \frac{\partial}{\partial \mathbf{n}} (\mathbf{u}_{n,j} \cdot \Gamma_n \cdot \mathbf{u}_{n,j}), \quad j = 1, 2, 3$$

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 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  $\eta_X$  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

$$\begin{aligned}\tilde{c}_E &= (1 + i\eta_{c_E})c_E \\ \tilde{s}_E &= (1 - i\eta_{s_E})s_E \\ \tilde{\epsilon}_{rS} &= (1 - i\eta_{\epsilon_{rS}})\epsilon_{rS} \\ \tilde{\epsilon}_{rT} &= (1 - i\eta_{\epsilon_{rT}})\epsilon_{rT} \\ \tilde{e} &= (1 + i\eta_e)e \\ \tilde{d} &= (1 + i\eta_d)d\end{aligned}$$

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  $\eta_X$  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  $\eta_e$  and  $\eta_d$  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 3-72 shows that  $e$  must have a positive loss factor.
- Consider a material with only coupling loss. Then the second transformation law in Equation 3-72 shows that  $d$  and  $e$  must have the same loss factor multiplier. From the third transformation law it can be inferred that  $\eta_e$  and  $\eta_d$  in this case must be positive in order for the material to have an appropriate phase lag between  $\mathbf{D}$  and  $\mathbf{E}$ .

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 electrical conductivity of the piezoelectric material,  $\sigma$ . Electrical 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 ( $\epsilon_0$ ), and remanent electric displacement ( $\mathbf{D}_r$ ) for models. In the constitutive relation for piezoelectric material these additions appear in the stress-charge formulation:

$$\begin{aligned}\boldsymbol{\sigma} &= c_E(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) - e^T \mathbf{E} + \boldsymbol{\sigma}_0 \\ \mathbf{D} &= e(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_0) + \epsilon_{0, \text{vac}} \boldsymbol{\epsilon}_{rS} \mathbf{E} + \mathbf{D}_r\end{aligned}$$



When solving the model, the program does not interpret these fields as a constant initial state, but they operate as additional fields that are continuously evaluated. You can thus use these initial field to add, for example, thermal expansion or pyroelectric effects to models.

## *Magnetostrictive Material*

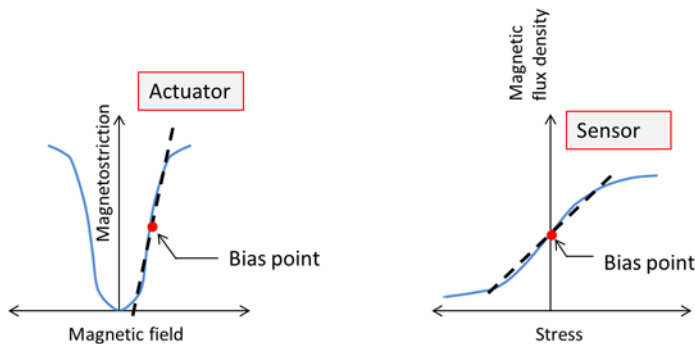
---

### **MAGNETOSTRICTION**

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.



### LINEAR MAGNETOSTRICTION MODEL

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).

It is possible to express the relation between the stress  $\mathbf{S}$ , strain  $\boldsymbol{\varepsilon}$ , magnetic field  $\mathbf{H}$ , and magnetic flux density  $\mathbf{B}$  in either a *stress-magnetization form* or *strain-magnetization form*:

*Stress-Magnetization*

$$\begin{aligned}\mathbf{S} &= c_{\mathbf{H}}\boldsymbol{\varepsilon} - e_{\mathbf{HS}}^T \mathbf{H} \\ \mathbf{B} &= e_{\mathbf{HS}}\boldsymbol{\varepsilon} + \mu_0 \mu_{\mathbf{rS}} \mathbf{H}\end{aligned}$$

*Strain-Magnetization*

$$\begin{aligned}\boldsymbol{\varepsilon} &= s_{\mathbf{H}}\mathbf{S} + d_{\mathbf{HT}}^T \mathbf{H} \\ \mathbf{B} &= d_{\mathbf{HT}}\mathbf{S} + \mu_0 \mu_{\mathbf{rT}} \mathbf{H}\end{aligned}$$

where  $\mu_0$  is the magnetic permeability of free space;  $c_{\mathbf{H}}$  and  $s_{\mathbf{H}}$  are the stiffness and compliance matrices measured at constant magnetic field, respectively; and  $\mu_{\mathbf{rS}}$  and  $\mu_{\mathbf{rT}}$  are the relative magnetic permeabilities measured at constant strain and constant stress, respectively. The matrices  $d_{\mathbf{HT}}$  and  $e_{\mathbf{HS}}$  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:

$$\begin{aligned}c_{\mathbf{H}} &= s_{\mathbf{H}}^{-1} \\ e_{\mathbf{HS}} &= d_{\mathbf{HT}} s_{\mathbf{H}}^{-1} \\ \mu_{\mathbf{rS}} &= \mu_{\mathbf{rT}} - \frac{1}{\mu_0} d_{\mathbf{HT}} s_{\mathbf{H}}^{-1} d_{\mathbf{HT}}^T\end{aligned}$$

One can rewrite the system of constitutive relations in the following equivalent form:

$$\begin{aligned}\mathbf{S} &= c_{\mathbf{H}}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathbf{me}}) \\ \mathbf{B} &= \mu_0 \mathbf{H} + \mu_0 (\mathbf{M} + \mu_0^{-1} d_{\mathbf{HT}}^T \mathbf{S})\end{aligned}$$

where the magnetostrictive strain is introduced as

$$\varepsilon_{me} = d_{HT}^T \mathbf{H}$$

and the material magnetization due to the applied field is given by

$$\mathbf{M} = (\mu_{rT} - I) \mathbf{H}$$

Hence, the stress in material is proportional to the elastic strain

$$\varepsilon_{el} = \varepsilon - \varepsilon_{me}$$

and it contributes to the material magnetization.

You find all the necessary material data inputs within the [Magnetostrictive Material](#) node under the Solid Mechanics interface, which are added automatically when you add a predefined Magnetostriction multiphysics interface. Such a node can be also added manually under any Solid Mechanics interface similar to all other material model features. The [Magnetostrictive 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. 6):

$$s_H = 10^{-11} \cdot \begin{bmatrix} 4.4 & -1.1 & -1.65 & 0 & 0 & 0 \\ -1.1 & 4.4 & -1.65 & 0 & 0 & 0 \\ -1.65 & -1.65 & 3.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 \end{bmatrix} [\text{1/Pa}]$$

$$d_{HT} = 10^{-9} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 16.5 & 0 \\ 0 & 0 & 0 & 16.5 & 0 & 0 \\ -4.3 & -4.3 & 8.6 & 0 & 0 & 0 \end{bmatrix} [\text{m/A}] \quad \mu_{rT} = \begin{bmatrix} 8.1 & 0 & 0 \\ 0 & 8.1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

### NONLINEAR MAGNETOSTRICTION

A commonly accepted micromagnetic description of the magnetostriction is as follows (Ref. 2):

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:

$$\varepsilon_{me} = \frac{3}{2} \left[ \lambda_{100} \left( \mathbf{m} \otimes \mathbf{m} - \frac{1}{3} \mathbf{I} \right) + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} m_i m_j (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \quad (3-75)$$

where  $(\mathbf{m} \otimes \mathbf{m})_{ij} = m_i m_j$ . Note that the magnetostrictive strain is represented by a deviatoric tensor, that is,  $\text{tr}(\varepsilon_{me}) = 0$ . 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  $\beta_i$  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 3-75, one gets

$$\lambda = \frac{3}{2} \lambda_{100} \left( m_1^2 \beta_1^2 + m_2^2 \beta_2^2 + m_3^2 \beta_3^2 - \frac{1}{3} \right) + 3 \lambda_{111} (m_1 m_2 \beta_1 \beta_2 + m_2 m_3 \beta_2 \beta_3 + m_1 m_3 \beta_1 \beta_3) \quad (3-76)$$

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

$$m_i = \beta_i = 1/\sqrt{3} \text{ and } \lambda = \lambda_{111}.$$

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 3-75 becomes

$$\varepsilon_{me} = \frac{3}{2} \lambda_s \left( \mathbf{m} \otimes \mathbf{m} - \frac{1}{3} I \right)$$

For a polycrystalline material without preferred orientation, the following approximation can be used (Ref. 1):

$$\lambda_s = \frac{2}{5}\lambda_{100} + \frac{3}{5}\lambda_{111}$$

In COMSOL Multiphysics, this description of the magnetostriction is modeled using the following equation for the magnetostrictive strain:

$$\epsilon_{me} = \frac{3}{2} \frac{1}{M_s^2} \left[ \lambda_{100} \text{dev}(\mathbf{M} \otimes \mathbf{M}) + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} M_i M_j (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \quad (3-77)$$

The strain field is deviatoric, and Equation 3-77 exhibits the same properties as Equation 3-75 at saturation, that is, when  $|\mathbf{M}| \rightarrow M_s$ . Equation 3-76 is replaced by

$$\begin{aligned} \lambda = & \frac{3\lambda_{100}}{2} \frac{1}{M_s^2} \left[ M_1^2 \left( \beta_1^2 - \frac{1}{3} \right) + M_2^2 \left( \beta_2^2 - \frac{1}{3} \right) + M_3^2 \left( \beta_3^2 - \frac{1}{3} \right) \right] + \\ & 3 \frac{\lambda_{111}}{M_s^2} (M_1 M_2 \beta_1 \beta_2 + M_2 M_3 \beta_2 \beta_3 + M_1 M_3 \beta_1 \beta_3) \end{aligned}$$

Note that the strain vanishes when  $|\mathbf{M}| \rightarrow 0$ , 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 (Ref. 3):

$$\epsilon_{me} = \frac{3}{2} \frac{\lambda_s}{M_s^2} \text{dev}(\mathbf{M} \otimes \mathbf{M}) \quad (3-78)$$

The stress in the magnetostrictive material is modeled as

$$\mathbf{S} = c_H [\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{me}(\mathbf{M})]$$

For isotropic materials, the stiffness matrix  $c_H$  can be represented in terms of two parameters, for example, using the Young's modulus and Poisson's ration. Cubic materials possess only three independent components:  $c_{11}$ ,  $c_{12}$ , and  $c_{44}$ .

Using Equation 3-78, one can derive a linear response around a given bias state characterized by a premagnetization vector  $\mathbf{M}_0$ . Thus,

$$\mathbf{M} = \mathbf{M}_0 + \mathbf{M}_1 = [M_{0,1}, M_{0,2}, M_{0,3}] + \mathbf{M}_1$$

where  $\mathbf{M}_1$  is a perturbation, and

$$\epsilon_{\text{me}} = \frac{\lambda_s}{M_s^2} \begin{bmatrix} 2M_{0,1} & -M_{0,2} & -M_{0,3} \\ -M_{0,1} & 2M_{0,2} & -M_{0,3} \\ -M_{0,1} & -M_{0,2} & 2M_{0,3} \\ 0 & 3M_{0,3} & 3M_{0,2} \\ 3M_{0,3} & 0 & 3M_{0,1} \\ 3M_{0,2} & 3M_{0,1} & 0 \end{bmatrix} \mathbf{M}_1$$

If one assumes a unidirectional state, for example

$$\mathbf{M}_0 = [0, 0, M_0]$$

it will further simplify into

$$\epsilon_{\text{me}} = \frac{\lambda_s M_0}{M_s^2} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{M}_1$$

### MAGNETIZATION

The magnetization in the magnetostrictive material is found from the following nonlinear implicit relation (Ref. 4 and Ref. 5):

$$\mathbf{M} = M_s L(|\mathbf{H}_{\text{eff}}|) \frac{\mathbf{H}_{\text{eff}}}{|\mathbf{H}_{\text{eff}}|} \quad (3-79)$$

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  $\chi_0$  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

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3}{\mu_0 M_s^2} \left[ \lambda_{100} \mathbf{S}_d + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} (S_d)_{ij} (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \mathbf{M} \quad (3-80)$$

where  $\mathbf{H}$  is the applied magnetic field. The second term in Equation 3-80 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

$$\mathbf{S}_d = \text{dev}(c_{\mathbf{H}} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{me}}))$$

For isotropic materials, the effective magnetic field expression simplifies into

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3\lambda_s}{\mu_0 M_s^2} \mathbf{S}_d \mathbf{M} \quad (3-81)$$

In addition, the magnetization and magnetic field are related to each other and to the magnetic flux density (also called the  $\mathbf{B}$ -field) by

$$\mathbf{B} = \mu_0 (\mathbf{H} + \mathbf{M})$$

COMSOL Multiphysics solves for the magnetic vector potential  $\mathbf{A}$  whose curl yields the vector  $\mathbf{B}$ -field. The  $\mathbf{H}$ -field is then obtained as a function of the  $\mathbf{B}$ -field and magnetization

$$\mathbf{H} = \mathbf{B}/\mu_0 - \mathbf{M}$$



The effective tangent piezomagnetic coupling coefficients can be computed as

$$\mathbf{d} = \frac{\partial \varepsilon_{me}}{\partial \mathbf{H}}$$

For an isotropic material, the derivative can be evaluated to give

$$\mathbf{d} = 3 \frac{\lambda_s}{M_s^2} \left( \mathbf{M} \otimes \chi_m - \frac{1}{3} (I \otimes I) : (\mathbf{M} \otimes \chi_m) \right)$$

where

$$\chi_m = \frac{\partial \mathbf{M}}{\partial \mathbf{H}}$$

is the tangent magnetic susceptibility. 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  $\mathbf{M}$  is zero at zero applied field, while  $\chi_m$  will 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  $d_{HTT}$  with only few non-zero components.

### HYSTERSIS 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 3-79](#), thus

$$\mathbf{M}_{an} = M_s L(|\mathbf{H}_{eff}|) \frac{\mathbf{H}_{eff}}{|\mathbf{H}_{eff}|}$$

Compared to [Equation 3-80](#) and [Equation 3-81](#), the effective magnetic field  $\mathbf{H}_{eff}$  gets one more term  $\alpha \mathbf{M}$ , where  $\alpha$  is the interdomain coupling parameter.

The change in the total magnetization caused by the change on the effective magnetic field is represented as

$$d\mathbf{M} = c_r d\mathbf{M}_{\text{an}} + \max(\xi \cdot d\mathbf{H}_{\text{eff}}, 0) \xi / |\xi|$$

$$\xi = k_p^{-1} (\mathbf{M}_{\text{an}} - \mathbf{M})$$
(3-82)

where  $c_r$  is the reversibility parameter, and  $k_p$  is the pinning loss parameter.

Equation 3-82 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. 3):

$$\varepsilon_{\text{em}} = \mathbf{Q} : (\mathbf{P} \otimes \mathbf{P})$$

which 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, the 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:

$$\varepsilon_{\text{em}} = \mathbf{QM} : (\mathbf{E} \otimes \mathbf{E})$$

where

$$\mathbf{M} = \varepsilon_{0, \text{vac}}^2 \mathbf{Q} : (\chi \otimes \chi)$$

where  $\varepsilon_{0, \text{vac}}$  is the electric permittivity of free space, and  $\chi$  is the linear electric susceptibility tensor (measured at zero mechanical deformation).

The total polarization in the material can then be written as

$$\mathbf{P} = \varepsilon_{0, \text{vac}} \chi \mathbf{E} + 2(\mathbf{M} : \mathbf{S}) \mathbf{E} \quad (3-83)$$

where the mechanical stress in the material is computed assuming a mechanically linear material as

$$\mathbf{S} = \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{em}})$$

where  $\mathbf{C}$  is the fourth order elasticity tensor, and the strain tensor is given by

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

The last term in [Equation 3-83](#) 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

$$M_{ij} = (\varepsilon_{0, \text{vac}} \chi_0)^2 Q_{ij}$$

where  $\chi_0$  denotes the only independent component of the electric susceptibility tensor.

For isotropic materials, the fourth order elasticity tensor  $\mathbf{C}$  has only two independent components. The most common choice to represent those are by specifying the Young's modulus  $E_{\text{YM}}$  and Poisson's ratio  $\nu$ .

The following alternative electrostrictive parameters are defined in [Ref. 1](#) for a linear isotropic material:

$$\begin{aligned} \alpha_1 &= 2(\varepsilon_{0, \text{vac}} \chi)^2 E_{\text{YM}} \frac{(1-\nu)Q_{11} + 2\nu Q_{12}}{(1+\nu)(1-2\nu)} \\ \alpha_2 &= 2(\varepsilon_{0, \text{vac}} \chi)^2 E_{\text{YM}} \frac{\nu Q_{11} + Q_{12}}{(1+\nu)(1-2\nu)} \end{aligned}$$

One more commonly used alternative definition is that introduced in [Ref. 2](#). Using  $\alpha_1'$  and  $\alpha_2'$  for the constants used in [Ref. 2](#), one has  $\alpha_1' = \alpha_1 - \alpha_2$  and  $\alpha_2' = \alpha_2$ .

The SI units for different electrostrictive parameters are summarized in [Table 3-6](#).

TABLE 3-6: UNITS FOR ELECTROSTRICTIVE CONSTANTS.

MATERIAL PARAMETER	UNIT (SI)
$Q$	$\text{m}^2 / \text{V}^2$
$M$	$\text{m}^4 / \text{C}^2$
$a$	$\text{F} / \text{m}$

## 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.
3. 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.

## *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. 1](#)):

$$\varepsilon_{em} = \mathbf{Q} : (\mathbf{P} \otimes \mathbf{P})$$

which is quadratic in 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:

$$\mathbf{P}_{\text{an}} = P_s L(|\mathbf{E}_{\text{eff}}|) \frac{\mathbf{E}_{\text{eff}}}{|\mathbf{E}_{\text{eff}}|} \quad (3-84)$$

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  $\chi_0$ .

For strong fields, the polarization magnitude approaches the saturation value

$$\mathbf{P} \approx P_s \frac{\mathbf{E}_{\text{eff}}}{|\mathbf{E}_{\text{eff}}|}$$

Two possible choices are the Langevin function

$$L = \coth\left(\frac{3\chi_0 |\mathbf{E}_{\text{eff}}|}{P_s}\right) - \frac{P_s}{3\chi_0 |\mathbf{E}_{\text{eff}}|}$$

and a hyperbolic tangent (Ising spin model):

$$L = \tanh\left(\frac{\chi_0 |\mathbf{H}_{\text{eff}}|}{P_s}\right)$$

The effective electric field is given by

$$\mathbf{E}_{\text{eff}} = \mathbf{E} + \alpha \mathbf{P} + 2(\mathbf{S} : \mathbf{Q}) \mathbf{P} \quad (3-85)$$

where  $\mathbf{E}$  is the applied electric field,  $\alpha$  is a material parameter called the inter-domain coupling, and the mechanics stress is computed assuming mechanically linear material as

$$\mathbf{S} = \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{em}})$$

where  $\mathbf{C}$  is the fourth order elasticity tensor. The last term in Equation 3-85 represents the inverse electrostrictive effect.

The effective tangent piezoelectric coupling tensor can be computed as

$$\mathbf{d} = \frac{\partial \boldsymbol{\varepsilon}_{\text{em}}}{\partial \mathbf{E}} = 2\varepsilon_{0, \text{vac}} \mathbf{Q} : (\mathbf{P} \otimes \boldsymbol{\chi}_e)$$

where  $\varepsilon_{0, \text{vac}}$  is the electric permittivity of free space, and

$$\boldsymbol{\chi}_e = \frac{1}{\varepsilon_{0, \text{vac}}} \frac{\partial \mathbf{P}}{\partial \mathbf{E}}$$

is the tangent electric susceptibility matrix. The piezoelectric coupling tensor  $\mathbf{d}$  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  $\boldsymbol{\chi}_e$  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 in case of isotropic material:

$$\boldsymbol{\varepsilon}_{\text{em}} = \frac{3\lambda_s}{2P_s^2} \text{dev}(\mathbf{P} \otimes \mathbf{P})$$

where  $\lambda_s$  is the saturation electrostriction, and the deviatoric part can be computed for any matrix  $\mathbf{A}$  as

$$\text{dev}(\mathbf{A}) = \mathbf{A} - \frac{1}{3}(\mathbf{I} \otimes \mathbf{I}) : \mathbf{A}$$

The stress in Equation 3-85 is replaced then by

$$\mathbf{S}_d = \text{dev}(\mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{em}))$$

and tangent piezoelectric coupling tensor becomes

$$\mathbf{d} = 3 \frac{\varepsilon_{0, \text{vac}} \lambda_s}{M_s^2} \left( \mathbf{P} \otimes \boldsymbol{\chi} - \frac{1}{3} (\mathbf{I} \otimes \mathbf{I}) : (\mathbf{P} \otimes \boldsymbol{\chi}) \right)$$

### 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:

$$d\mathbf{P} = c_r d\mathbf{P}_{an} + (\mathbf{I} - c_r) d\mathbf{P}_{irr} \quad (3-86)$$

where the anhysteretic polarization is found using [Equation 3-84](#), and the irreversible polarization change is computed as

$$d\mathbf{P}_{irr} = \max(\zeta \cdot d\mathbf{E}_{eff}, 0) \frac{\zeta}{|\zeta|}$$

$$\zeta = \mathbf{k}_p^{-1} (\mathbf{P}_{an} - \mathbf{P}_{irr})$$

where the pinning loss is characterized by the parameter  $\mathbf{k}_p$ .

[Equation 3-86](#) can be solved using either a time-dependent analysis or a stationary parametric sweep.

### REFERENCES

1. 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.
2. C. L. Horn and N. Shankar, “A finite element method for electrostrictive ceramic devices,” *Int. J. Solids Structures*, vol. 33, pp. 1757–1779, 1995.
3. 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.

### *Rigid Domain Model*

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.



---

Rigid Domain is available as a material model both 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 Domain which will be described as they appear.

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 Domains nodes are independent, and can even penetrate each other.

---

The rigid domain 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, the 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 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 however 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 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  $\phi$ , 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:

$$\mathbf{u}_d = \mathbf{u} + (R - I) \cdot (\mathbf{X} - \mathbf{X}_M)$$

where  $\mathbf{X}$  are the material coordinates of any point in the rigid domain,  $\mathbf{X}_M$  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:

$$\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t} \quad \ddot{\mathbf{u}} = \frac{\partial}{\partial t} \dot{\mathbf{u}}$$

In the frequency domain, they can be expressed in terms of frequency ( $\omega$ ):

$$\dot{\mathbf{u}} = (i\omega)\mathbf{u} \quad \ddot{\mathbf{u}} = (i\omega)\dot{\mathbf{u}} = -\omega^2\mathbf{u}$$

The same is true for the rotation in 2D since the rigid body rotation  $\phi$  is the degree of freedom. The rigid body angular velocity and acceleration can be evaluated by directly taking the time derivatives of  $\phi$ .

In 3D, the situation is different and the total rotation of the rigid domain can be presented as a function of quaternion:

$$\Theta = \frac{2a \cos(a)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

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:

$$\dot{\Theta} = \begin{bmatrix} Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad Q = 2(\bar{q} \otimes \dot{q}) \quad q = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

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:

$$\mathbf{F} = m\ddot{\mathbf{u}} \quad \mathbf{M} = I\ddot{\Theta}$$

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:

$$m = \int \rho dV$$

$$\mathbf{X}_M = \frac{\int \rho \mathbf{X} dV}{m}$$

$$I = \int ((\mathbf{X} - \mathbf{X}_M)^T \cdot (\mathbf{X} - \mathbf{X}_M)) E_3 - (\mathbf{X} - \mathbf{X}_M) \cdot (\mathbf{X} - \mathbf{X}_M)^T \rho dV$$

where  $E_3$  and  $\mathbf{X}_M$  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:

$$\mathbf{F} = m\ddot{\mathbf{u}} \quad M = I_z \ddot{\phi}$$

$$I_z = \int ((\mathbf{X} - \mathbf{X}_M) \cdot (\mathbf{X} - \mathbf{X}_M)) \rho h dA$$

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:

$$m\ddot{\mathbf{u}} + \sum \mathbf{F}_I = \sum \mathbf{F}_{\text{ext}}$$

and

$$RIR^T\ddot{\Theta} + \dot{\Theta} \times (RIR^T\dot{\Theta}) + \sum \mathbf{M}_I = \sum \mathbf{M}_{\text{ext}}$$

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

$$I_z\ddot{\phi} + \sum M_I = \sum M_{\text{ext}}$$

### INITIAL VALUE

As a Rigid Domain is a separate material model, it overrides the default Linear Elastic Material model and its default Initial Values node. The initial values are given in a separate Initial Values subnode for each Rigid Domain.



In the Multibody Dynamics interface version of the Rigid Domain, 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 domain 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 it 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 ( $\phi$ ), 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:

$$\begin{aligned}\mathbf{u}_0 &= \mathbf{u} + \mathbf{u}_r \\ \mathbf{u}_r &= (-\mathbf{r} + \hat{\Omega}(\hat{\Omega} \cdot \mathbf{r}))(1 - \cos \phi) + (\hat{\Omega} \times \mathbf{r}) \sin \phi \\ \mathbf{r} &= \mathbf{X}_M - \mathbf{X}_c \\ \left(\frac{\partial \mathbf{u}}{\partial t}\right)_0 &= (\omega \times (\mathbf{r} + \mathbf{u}_r)) + \frac{\partial \mathbf{u}}{\partial t} \\ \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0 &= \begin{bmatrix} \cos \frac{\phi}{2} \\ \sin \frac{\phi}{2} \hat{\Omega} \end{bmatrix} \\ \begin{bmatrix} \frac{\partial a}{\partial t} \\ \frac{\partial \mathbf{b}}{\partial t} \end{bmatrix}_0 &= \frac{1}{2} \left( \begin{bmatrix} 0 \\ \omega \end{bmatrix} \otimes \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0 \right)\end{aligned}$$

In 2D, the expressions for the initial values reduce to:

$$\begin{aligned}\mathbf{u}_0 &= \mathbf{u} + \mathbf{u}_r \\ \mathbf{u}_r &= -\mathbf{r}(1 - \cos \phi) + (\mathbf{e}_z \times \mathbf{r}) \sin \phi \\ \mathbf{r} &= \mathbf{X}_M - \mathbf{X}_c \\ \left(\frac{\partial \mathbf{u}}{\partial t}\right)_0 &= (\mathbf{e}_z \times (\mathbf{r} + \mathbf{u}_r)) \frac{\partial \phi}{\partial t} + \frac{\partial \mathbf{u}}{\partial t} \\ \phi_0 &= \phi \\ \left(\frac{\partial \phi}{\partial t}\right)_0 &= \frac{\partial \phi}{\partial t}\end{aligned}$$

The variable  $\mathbf{u}_r$  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.

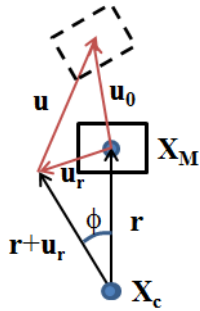


Figure 3-27: Initial displacement of a rigid body

### 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

$$\mathbf{F}_I = m \frac{d^2}{dt^2} (\mathbf{u} + (R - E_3) \mathbf{X}_{mc})$$

where  $\mathbf{X}_{mc}$  is the vector from the center of mass of the rigid domain ( $\mathbf{X}_M$ ) to the center of mass of this contribution ( $\mathbf{X}_m$ ),

$$\mathbf{X}_{mc} = \mathbf{X}_m - \mathbf{X}_M$$

The inertial moment contributions are

$$\mathbf{M}_I = RIR^T \ddot{\Theta} + \dot{\Theta} \times (RIR^T \dot{\Theta}) + \mathbf{X}_{mc} \times \mathbf{F}_I$$

In 2D, there is only a scalar moment contribution:

$$M_I = I_z \ddot{\phi} + (\mathbf{X}_{mc} \times \mathbf{F}_I) \cdot \mathbf{e}_z$$

### CONSTRAINTS

The constraints for a rigid domain are different in nature than those applied to flexible domain. 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:

$$\mathbf{u}_c = \mathbf{u} + (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X}_c - \mathbf{X}_M) \quad (3-87)$$

The components of this displacement vector are prescribed individually in the selected coordinate system. Through [Equation 3-87](#), 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  $\phi_0$  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

$$\mathbf{u}_{\text{flex}} = (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X} - \mathbf{X}_M) + \mathbf{u}$$



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.

A Rigid Domain which contains a selection of several geometrical domains will act as a single rigid object.

Adjacent domains selected in different Rigid Domains nodes are independent objects.

### *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  $\sigma_{un}$  to the elastic strain tensor:

$$\sigma_{un} = \sigma_{ex} + \mathbf{C} : \varepsilon_{el} = \sigma_{ex} + \mathbf{C} : (\varepsilon - \varepsilon_{inel}) \quad (3-88)$$

here,  $\mathbf{C}$  is the fourth order *elasticity tensor*, “:” stands for the double-dot tensor product (or double contraction). The elastic strain  $\varepsilon_{el}$  is the difference between the total strain  $\varepsilon$  and inelastic strains  $\varepsilon_{inel}$ . There may also be an external stress contribution  $\sigma_{ex}$ , with contributions from initial, viscoelastic, or other inelastic stresses.

For the scalar damage models, the *damaged* stress tensor  $\sigma_d$  is computed from the undamaged stress as

$$\sigma_d = (1 - d)\sigma_{un} \quad (3-89)$$

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.



The undamaged stress tensor  $\sigma_{un}$  is used when combining the **Damage** feature with [Creep and Viscoplasticity](#) or [Linear Viscoelastic Materials](#).

### 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

$$f = \varepsilon_{eq} - \kappa \leq 0 \quad (3-90)$$

here,  $\varepsilon_{eq}$  is the equivalent strain, a scalar measure of the elastic strain; and  $\kappa$  is a state variable. The evolution of the state variable  $\kappa$  follows the Kuhn–Tucker loading/unloading conditions

$$f \leq 0, \quad \dot{\kappa} \geq 0, \quad \text{and} \quad \dot{\kappa}f = 0$$

In this formulation,  $\kappa$  is the maximum value of  $\varepsilon_{eq}$  in the load history. The damage variable  $d$  is then computed as a function of the state variable  $\kappa$  and other parameters.

#### *Equivalent Strain*

Different damage models use different definitions for the equivalent strain  $\varepsilon_{eq}$ . The **Rankine, stress** damage model defines the equivalent strain from the largest undamaged principal stress  $\sigma_{p1}$  as

$$\varepsilon_{eq} = \frac{\langle \sigma_{p1} \rangle}{E} \quad (3-91)$$

here, the symbol “ $\langle \rangle$ ” 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  $\varepsilon_{elp1}$  as

$$\varepsilon_{\text{eq}} = \langle \varepsilon_{\text{clp1}} \rangle \quad (3-92)$$

The **Smooth Rankine, stress** damage model defines the equivalent strain from the three undamaged principal stresses

$$\varepsilon_{\text{eq}} = \frac{\sqrt{\langle \sigma_{\text{p1}} \rangle^2 + \langle \sigma_{\text{p2}} \rangle^2 + \langle \sigma_{\text{p3}} \rangle^2}}{E} \quad (3-93)$$

In the same way, the **Smooth Rankine, strain** damage model defines the equivalent strain from the three elastic principal strains

$$\varepsilon_{\text{eq}} = \sqrt{\langle \varepsilon_{\text{clp1}} \rangle^2 + \langle \varepsilon_{\text{clp2}} \rangle^2 + \langle \varepsilon_{\text{clp3}} \rangle^2} \quad (3-94)$$

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  $\sigma_{\text{d}}$ .

The Euclidean **Norm of the elastic strain tensor** can also be used as a measure for the equivalent strain

$$\varepsilon_{\text{eq}} = \sqrt{\varepsilon_{\text{cl}} : \varepsilon_{\text{cl}}} \quad (3-95)$$

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

$$\varepsilon_{\text{eq}} = \sqrt{\langle \varepsilon_{\text{cl}} \rangle : \langle \varepsilon_{\text{cl}} \rangle} \quad (3-96)$$

In the modified Mazars equivalent strain ([Ref. 2](#) and [Ref. 3](#)), a correction factor  $\gamma$  is added to improve the approximation of the failure surface of concrete in multiaxial compression.

$$\varepsilon_{\text{eq}} = \gamma \sqrt{\langle \varepsilon_{\text{cl}} \rangle : \langle \varepsilon_{\text{cl}} \rangle} \quad (3-97)$$

It is also possible to apply 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

$$\begin{aligned}
d(\kappa) &= \left(1 - \frac{\varepsilon_0}{\kappa}\right) \left(\frac{\varepsilon_f}{\varepsilon_f - \varepsilon_0}\right) & \kappa \geq \varepsilon_0, \\
d(\kappa) &= 0 & \kappa < \varepsilon_0
\end{aligned} \tag{3-98}$$

Here,  $\varepsilon_0$  denotes the onset of damage, computed from the tensile strength  $\sigma_{ts}$  and Young's modulus  $E$ , so that  $\varepsilon_0 = \sigma_{ts}/E$ . The parameter  $\varepsilon_f$  is derived from other material parameters such as the tensile strength, the characteristic element size  $h_{cb}$  and the fracture energy per unit area  $G_f$ , or the fracture energy per unit volume  $g_f$ .

$$\varepsilon_f = \frac{2G_f}{\sigma_{ts}h_{cb}} + \frac{\varepsilon_0}{2} \tag{3-99}$$

The **Exponential strain softening** law defines the damage evolution from

$$\begin{aligned}
d(\kappa) &= 1 - \frac{\varepsilon_0}{\kappa} \exp\left(-\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0}\right) & \kappa \geq \varepsilon_0 \\
d(\kappa) &= 0 & \kappa < \varepsilon_0
\end{aligned} \tag{3-100}$$

where

$$\varepsilon_f = \frac{G_f}{\sigma_{ts}h_{cb}} + \frac{\varepsilon_0}{2} \tag{3-101}$$

The **Polynomial strain softening** law defines the damage evolutions from

$$\begin{aligned}
d(\kappa) &= 1 - \frac{\varepsilon_0}{\kappa} \left[1 + \left(\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0}\right)^2 \left(2\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0} - 3\right)\right] & \kappa \geq \varepsilon_0 \\
d(\kappa) &= 0 & \kappa < \varepsilon_0
\end{aligned} \tag{3-102}$$

with  $\varepsilon_f$  according to [Equation 3-99](#).

The **Multilinear strain softening** law defines the damage evolutions from

$$\begin{aligned}
d(\kappa) &= 1 - \frac{\varepsilon_0}{\kappa} \frac{\varepsilon_f - \kappa}{\varepsilon_f - \varepsilon_p} & \kappa \geq \varepsilon_p \\
d(\kappa) &= 1 - \frac{\varepsilon_0}{\kappa} & \varepsilon_0 < \kappa < \varepsilon_p \\
d(\kappa) &= 0 & \kappa < \varepsilon_0
\end{aligned} \tag{3-103}$$

with  $\varepsilon_f$  according to Equation 3-99 and

$$\varepsilon_p = \lambda \frac{G_f}{\sigma_{ts} h_{cb}} + \varepsilon_0 \left(1 - \frac{\lambda}{2}\right) \quad (3-104)$$

where  $\lambda$  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

$$d(\kappa) = \alpha_t^\beta d_t(\kappa) + \alpha_c^\beta d_c(\kappa) \quad (3-105)$$

here,  $\alpha_t$  and  $\alpha_c$  are weight functions depending on the current stress state, and  $\beta$  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. 3), 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

$$\begin{aligned} d_t(\kappa) &= 1 - (1 - A_t) \frac{\varepsilon_{0t}}{\kappa} - A_t \exp(-B_t(\kappa - \varepsilon_{0t})) & \kappa \geq \varepsilon_{0t} \\ d_t(\kappa) &= 0 & \kappa < \varepsilon_{0t} \end{aligned} \quad (3-106)$$

Here,  $A_t$  and  $B_t$  are tensile damage evolution parameters, and  $\varepsilon_{0t}$  is the tensile strain threshold.

The compressive *Mazars damage evolution function*  $d_c(\kappa)$  is obtained by

$$\begin{aligned} d_c(\kappa) &= 1 - (1 - A_c) \frac{\varepsilon_{0c}}{\kappa} - A_c \exp(-B_c(\kappa - \varepsilon_{0c})) & \kappa \geq \varepsilon_{0c} \\ d_c(\kappa) &= 0 & \kappa < \varepsilon_{0c} \end{aligned} \quad (3-107)$$

Here,  $A_c$  and  $B_c$  are compressive damage evolution parameters, and  $\varepsilon_{0c}$  is the compressive strain threshold. Both the tensile and compressive damage evolution laws can also be specified by **User defined** expressions.

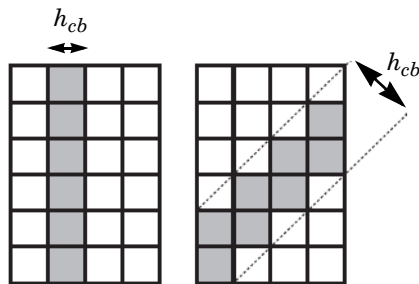
### Spatial Regularization

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. 4, Ref. 5). 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:

- $h_{cb} = \sqrt{2a}$ , for 2D triangles
- $h_{cb} = \sqrt{a}$ , for 2D rectangles
- $h_{cb} = \sqrt[3]{6/\sqrt{2v}}$ , for 3D tetrahedra
- $h_{cb} = \sqrt[3]{6v/\sqrt{2}}$ , for 3D pyramids
- $h_{cb} = \sqrt[3]{2v}$ , for 3D wedges
- $h_{cb} = \sqrt[3]{v}$ , 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 3-106](#) and [Equation 3-107](#)) are unaffected by the crack band method.

#### *The Implicit Gradient Method*

The *Implicit Gradient* method ([Ref. 6](#)) 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*  $\varepsilon_{nl}$ , through an additional PDE where the equivalent strain  $\varepsilon_{eq}$  acts as source term. This PDE is solved simultaneously with the displacement field:

$$\varepsilon_{nl} - c \nabla^2 \varepsilon_{nl} = \varepsilon_{eq} \quad (3-108)$$

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)

$$c = \frac{l_{int}^2}{2n} \quad (3-109)$$

If the fracture energy per unit 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 3-90](#)) is then redefined by the nonlocal equivalent strain  $\varepsilon_{nl}$  instead of the equivalent strain  $\varepsilon_{eq}$

$$f = \varepsilon_{nl} - \kappa \leq 0 \quad (3-110)$$

### 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

$$\tau \frac{\partial d_v}{\partial t} + d_v = d(\kappa) \quad (3-111)$$

where  $\tau$  is the characteristic time. For the case when the viscous regularization is used to stabilize a rate-independent fracture problem,  $\tau$  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 3-89](#) 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. 7](#), 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. Including a viscous regularization of the crack phase field  $\phi$ , the energy functional governing crack propagation takes the following form:

$$\begin{aligned} E(\mathbf{x}, t, \mathbf{u}, \nabla \mathbf{u}, \phi, \nabla \phi) = & \int_{\Omega} [1 - d(\phi)] W_{s0} d\mathbf{x} + \int_{\Omega} \eta \frac{\partial \phi}{\partial t} d\mathbf{x} \\ & + \int_{\Omega} G_c \left( \frac{1}{2l_{\text{int}}} \phi^2 + \frac{l_{\text{int}}}{2} |\nabla \phi|^2 \right) d\mathbf{x} \end{aligned} \quad (3-112)$$

where  $d(\phi)$  is the damage function,  $W_{s0}$  is the elastic strain energy density,  $\eta$  a viscosity,  $G_c$  is the critical energy release rate, and  $l_{\text{int}}$  is the internal length scale that appears from the regularization of the discrete fracture. Given the two dependent variables  $\mathbf{u}$  and  $\phi$ , the main contributions to the coupled initial boundary value problem, including inertia effects, are:



$$\left\{ \begin{array}{ll} \rho \frac{d^2 \mathbf{u}}{dt^2} = \nabla \cdot \boldsymbol{\sigma}_d + \mathbf{F}_V & \text{on } \Omega \\ \tau \frac{\partial \phi}{\partial t} = \frac{\partial d(\phi)}{\partial t} H_d - \phi - l_{\text{int}}^2 \nabla^2 \phi & \text{on } \Omega \\ 0 = -\nabla \phi \cdot \mathbf{n} & \text{on } \partial \Omega \end{array} \right. \quad (3-113)$$

where the characteristic time  $\tau$  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:

$$H_d(t, \mathbf{u}) = \max_{\tau \in [0, t]} D_d(\tau, \mathbf{u}) \quad (3-114)$$

The momentum balance in Equation 3-113 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 3-89, is used for the mechanical equilibrium.

#### Crack Driving Force

The crack driving force is typically a function of the elastic strain energy density, and by choosing **Strain energy density** it is defined as

$$D_d = \frac{l_{\text{int}}}{G_c - G_{c0}} \langle W_{s0}^+ - \frac{G_{c0}}{l_{\text{int}}} \rangle \quad (3-115)$$

where  $W_{s0}^+$  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 domain before damage occurs. By setting  $G_{c0}$  to zero the model reported in Ref. 7 is recovered.

An alternative definition of the strain energy density  $D_d$  based on a stress measure is suggested in Ref. 8. By selecting a **Principal stress criterion**, the crack driving force is defined as

$$D_d = \xi \left\langle \sum_{i=1}^3 \left( \frac{\sigma_{pi}}{\sigma_c} \right)^2 \right\rangle \quad (3-116)$$

where  $\sigma_c$  is the *critical fracture stress*,  $\sigma_{pi}$  are the *undamaged principal stresses*, and  $\xi$  is a dimensionless parameter which determines the post peak slope.

### Damage Evolution for Phase Field Damage

The damage variable  $d$  is in the phase field damage model a function of the crack phase field  $\phi$ . By selecting **Power law**, the damage evolution function is defined as

$$d(\phi) = 1 - (1 - \phi)^m \quad (3-117)$$

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. 9](#), and by selecting **Cubic, Borden**, the damage evolution function  $d(\phi)$  reads

$$d(\phi) = 1 - s[(1 - \phi)^3 - (1 - \phi)^2] - 3(1 - \phi)^2 + 2(1 - \phi)^3 \quad (3-118)$$

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.

### 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. This alternative is used by selecting **No** in the **Exclude compressive energy** list. This option has the advantage that it greatly simplifies the structure of the equations, although that comes at a cost of physical correctness, especially for shear and cyclic loading.

A more physically correct 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  $\sigma_{\text{dev}} = \text{dev}(\sigma_{\text{un}})$  is included in  $W_{s0}^+$ . The definition of  $W_{s0}^+$  then reads

$$W_{s0}^+ = \frac{\langle I_1 \rangle \epsilon_{\text{el,vol}}}{6} + \frac{1}{2} \sigma_{\text{dev}} : \epsilon_{\text{el,dev}} \quad (3-119)$$

where  $I_1$  is the first invariant of the undamaged stress tensor defined in [Equation 3-88](#) (see [Invariants of the Stress Tensor](#)); and  $\epsilon_{\text{el,vol}}$  and  $\epsilon_{\text{el,dev}}$  are the volumetric and deviatoric parts of the elastic strain tensor, respectively (see [Invariants of Strain](#)).

Another alternative is to define  $W_{s0}^+$  by making a spectral decomposition of the stress tensor such that

$$W_{s0}^+ = \frac{1}{2} \sum_{i=1}^3 \langle \sigma_{pi} \rangle \epsilon_{el, pi} \quad (3-120)$$

where  $\sigma_{pi}$  are the principal values of the undamaged stress tensor, and  $\epsilon_{el, pi}$  are the principal values of the elastic strain tensor (see [Principal Strains](#)).

Similarly, a spectral decomposition can be made directly on the elastic strain tensor as suggested in [Ref. 7](#). The tensile part of the elastic strain tensor is then

$$\epsilon_{el}^+ = \sum_{i=1}^3 \langle \epsilon_{el, pi} \rangle \mathbf{n}_i \otimes \mathbf{n}_i \quad (3-121)$$

and the tensile part of the undamaged stress tensor is defined as

$$\sigma^+ = \mathbf{C} : \epsilon_{el}^+ \quad (3-122)$$

Using these two tensors, the tensile part of the elastic strain energy density is

$$W_{s0}^+ = \frac{1}{2} \sigma^+ : \epsilon_{el}^+ \quad (3-123)$$

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 model, but they are the most correct formulations to use.

The choice of how to split the elastic strain energy density also affects how the damaged stress tensor is constructed from the tensile part of the undamaged stress tensor.

### *Safety Factor Evaluation*

---

There are many theories available in the literature 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. 7](#), [Ref. 16](#)) 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

$$g(\sigma) = \sigma : (F\sigma) + f : \sigma - 1$$

here,  $\sigma$  is the stress tensor,  $F$  a fourth rank tensor (SI unit:  $1/\text{Pa}^2$ ) and  $f$  is a second rank tensor (SI unit:  $1/\text{Pa}$ ). 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 to be positive only. These restrictions are summarized as (no summation of the indices)

$$F_{ii} > 0 \quad \text{and} \quad F_{ii}F_{jj} \geq F_{ij}^2$$

The *failure index* is computed from the failure criterion as

$$\text{fi} = g(\sigma) + 1$$

so failure is predicted for a failure index greater than one,  $\text{fi} \geq 1$ .

The *damage index* is given by a Boolean expression based on the failure criterion

$$\text{di} = \begin{cases} 1 & g(\sigma) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

here  $\text{di} = 1$  means damage, and  $\text{di} = 0$  represents a healthy material.

The *safety factor*, also called *reserve factor or strength ratio*, is computed by scaling the stress tensor such as the failure criterion is equal to zero

$$g(s_f \sigma) = 0$$

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,  $g(\sigma) = \sigma_{\text{mises}}/\sigma_{\text{ts}} - 1$ , and the safety factor is given by  $s_f = \sigma_{\text{ts}}/\sigma_{\text{mises}}$ .

The *margin of safety* ([Ref. 16](#)) is then computed from the safety factor

$$ms = s_f - 1$$

Following the Tsai–Wu formalism, different orthotropic criteria can be defined by setting appropriate values for the coefficients in  $F$  and  $f$  tensors.

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](#), [Layered Linear Elastic Material](#), or [Nonlinear Elastic Materials](#).

### THE 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,  $\sigma_{ij}$  are the stress tensor components given in the local coordinate system of the parent node.

### THE ORTHOTROPIC TSAI–WU CRITERION

For this orthotropic criterion, enter nine coefficients corresponding to the tensile strengths  $\sigma_{tsi}$ , compressive strengths  $\sigma_{csi}$ , and shear strengths  $\sigma_{ssij}$  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  $f$  tensors are set to zero.

#### THE TSAI–HILL CRITERION

For this orthotropic criterion, enter six coefficients corresponding to the tensile strengths  $\sigma_{tsi}$  and shear strengths  $\sigma_{ssij}$  given in the local coordinate system of the parent node. Tsai–Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{ts1}^2}, F_{22} = \frac{1}{\sigma_{ts2}^2}, F_{33} = \frac{1}{\sigma_{ts3}^2},$$

$$F_{44} = \frac{1}{\sigma_{ss23}^2}, F_{55} = \frac{1}{\sigma_{ss13}^2}, F_{66} = \frac{1}{\sigma_{ss12}^2},$$

$$F_{12} = -\frac{1}{2}(F_{11} + F_{22} - F_{33}), F_{13} = -\frac{1}{2}(F_{11} + F_{33} - F_{22}),$$

$$F_{23} = -\frac{1}{2}(F_{22} + F_{33} - F_{11})$$

all the other coefficients in  $F$  and  $f$  tensors are set to zero. See also [Hill Orthotropic Plasticity](#).

#### THE HOFFMAN CRITERION

For this orthotropic criterion, enter nine coefficients corresponding to the tensile strengths  $\sigma_{tsi}$ , compressive strengths  $\sigma_{csi}$ , and shear strengths  $\sigma_{ssij}$  given in the local coordinate system of the parent node. The Tsai–Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{cs1}\sigma_{ts1}}, F_{22} = \frac{1}{\sigma_{cs2}\sigma_{ts2}}, F_{33} = \frac{1}{\sigma_{cs3}\sigma_{ts3}},$$

$$F_{44} = \frac{1}{\sigma_{ss23}^2}, F_{55} = \frac{1}{\sigma_{ss13}^2}, F_{66} = \frac{1}{\sigma_{ss12}^2},$$

$$F_{12} = -\frac{1}{2}(F_{11} + F_{22} - F_{33}), F_{13} = -\frac{1}{2}(F_{11} + F_{33} - F_{22}),$$

$$F_{23} = -\frac{1}{2}(F_{22} + F_{33} - F_{11})$$

$$f_1 = \frac{1}{\sigma_{ts1}} - \frac{1}{\sigma_{cs1}}, f_2 = \frac{1}{\sigma_{ts2}} - \frac{1}{\sigma_{cs2}}, f_3 = \frac{1}{\sigma_{ts3}} - \frac{1}{\sigma_{cs3}}$$

all the other coefficients in  $F$  and  $f$  tensors are set to zero.

### THE JENKINS CRITERION

For Jenkins orthotropic criterion, enter nine coefficients corresponding to the tensile strengths  $\sigma_{t_{si}}$ , compressive strengths  $\sigma_{c_{si}}$ , and shear strengths  $\sigma_{ssij}$  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,  $\varepsilon_{si}$  is either the tensile strength or the compressive strength depending whether the stress in the  $i$  direction,  $\sigma_i$ , is positive or negative. The absolute value of the shear stress  $\sigma_{ij}$  in the  $ij$ -plane is compared to the corresponding shear strength  $\sigma_{ssij}$ .

### THE 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  $\varepsilon_{t_{si}}$ , ultimate compressive strains  $\varepsilon_{c_{si}}$ , and ultimate shear strains  $\gamma_{ssij}$  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,  $\varepsilon_{si}$  is either the ultimate tensile strain or the ultimate compressive strain depending whether the strain in the  $i$  direction,  $\varepsilon_i$ , is positive or negative. The absolute value of the shear strain  $\gamma_{ij}$  in the  $ij$ -plane is compared to the corresponding ultimate shear strain  $\gamma_{ssij}$ .

### THE MODIFIED TSAI-HILL CRITERION

This criterion is derived from the Tsai–Wu theory for two-dimensional plane stress problems (Ref. 17). 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  $\sigma_{t_{si}}$ , compressive strengths  $\sigma_{c_{si}}$ , and shear strengths  $\sigma_{ssij}$  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

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0, \text{ or } F_{11} = \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts2}^2} \quad \text{for } \sigma_{22} > 0, \text{ or } F_{22} = \frac{1}{\sigma_{cs2}^2} \quad \text{for } \sigma_{22} < 0$$

$$F_{12} = -\frac{1}{2} \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11}\sigma_{22} > 0 \text{ or } F_{12} = -\frac{1}{2} \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11}\sigma_{22} < 0$$

$$F_{66} = \frac{1}{\sigma_{ss12}^2}$$

all the other coefficients in  $F$  and  $f$  tensors are set to zero.

#### THE 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 and the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths  $\sigma_{tsi}$ , compressive strengths  $\sigma_{csi}$ , and shear strengths  $\sigma_{ssij}$  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 in  $F$  and  $f$  tensors are set to zero.



### THE 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  $\sigma_{tsi}$ , compressive strengths  $\sigma_{csi}$ , and shear strengths  $\sigma_{ssij}$  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 in  $F$  and  $f$  tensors are set to zero.

### THE 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  $\sigma_{ts}$

$$g(\sigma) = \frac{\sigma_{mises}}{\sigma_{ts}} - 1$$

The equivalent von Mises stress  $\sigma_{mises}$  is defined from the deviatoric stress tensor, see the section about plasticity and [The von Mises Criterion](#). For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

### THE 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  $\sigma_{ts}$

$$g(\sigma) = \frac{\sigma_{tresca}}{\sigma_{ts}} - 1$$

Here, the Tresca equivalent stress is defined in terms of principal stresses,  $\sigma_{tresca} = \sigma_1 - \sigma_3$ ; see [The Tresca Criterion](#). For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

### THE 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  $\sigma_{ts}$ , and the compressive strength  $\sigma_{cs}$ . The failure criterion is then computed from

$$g(\sigma) = \max\left(\frac{|\sigma_{p1}|}{\sigma_s}, \frac{|\sigma_{p2}|}{\sigma_s}, \frac{|\sigma_{p3}|}{\sigma_s}\right) - 1$$

here,  $\sigma_s$  is either the tensile strength or the compressive strength depending whether the principal stress,  $\sigma_{pi}$ , 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.

### THE ST. VENANT CRITERION

The St. 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  $\epsilon_{ts}$ , and the ultimate compressive strains  $\epsilon_{cs}$ . The failure criterion is then computed from

$$g(\epsilon) = \max\left(\frac{\epsilon_{p1}}{\epsilon_s}, \frac{\epsilon_{p2}}{\epsilon_s}, \frac{\epsilon_{p3}}{\epsilon_s}\right) - 1$$

here,  $\epsilon_s$  is either the ultimate tensile strain or the ultimate compressive strain depending whether the principal strain,  $\epsilon_{pi}$ , 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.

### THE 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 [The Mohr–Coulomb Criterion](#) for soil plasticity. For this isotropic criterion, enter the cohesion  $c$ , and the angle of internal friction  $\phi$ . 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

$$m(\theta) = \sqrt{\frac{1}{3}} \left( (1 + \sin\phi) \cos\theta - (1 - \sin\phi) \cos\left(\theta + \frac{2\pi}{3}\right) \right), \quad \alpha = \sin\phi/3, \text{ and}$$

$$k = c \cos\phi$$

The cohesion and the angle of internal friction are related to the tensile and compressive strengths by the expressions

$$\sin(\phi) = \frac{\sigma_{cs1} - \sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}} \quad \text{and} \quad c \cos(\phi) = \frac{\sigma_{cs1} \sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}}$$

### THE DRUCKER–PRAGER CRITERION

The Drucker–Prager criterion approximates the Mohr–Coulomb criterion by a smooth function (a cone in the stress space), see [The 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,  $\alpha$  and  $k$ ,

$$g(\sigma) = \frac{\sqrt{J_2}}{k - \alpha I_1} - 1$$

The material parameters  $\alpha$  and  $k$  are related to the cohesion  $c$  and angle of internal friction  $\phi$  in the Mohr–Coulomb criterion, see [The Drucker–Prager Criterion](#) for details. Also, the cohesion and the angle of internal friction can be related to the tensile and compressive strengths, see [The Mohr–Coulomb Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{k - \alpha I_1}$$

### THE 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 [The 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  $\sigma_c$ , the uniaxial tensile strength  $\sigma_t$ , and the biaxial compression strength  $\sigma_b$ , see [The Bresler–Pister Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{-k_1 I_1^2 - k_2 I_1 - k_3}$$

### THE 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 [The Bresler–Pister Criterion](#), failure is computed from the stress invariants  $I_1$  and  $J_2$ , and the Lode angle  $\theta$ , and three material parameters

$$g(\sigma) = \frac{\sqrt{J_2}}{\sqrt{\frac{5}{2}} r(\theta) \sigma_c \left( 1 + \left( \frac{1}{\sigma_b} - \frac{1}{\sigma_t} \right) \frac{I_1}{3} \right)} - 1$$

here,  $\sigma_c$  is the uniaxial compressive strength,  $\sigma_t$  is uniaxial tensile strength, and  $\sigma_b$  is the biaxial compressive strength. The function  $r(\theta)$  describes the segment of an ellipse on the octahedral plane, see [The Willam–Warnke Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{\sqrt{\frac{5}{2}} r(\theta) \sigma_c \left( 1 + \left( \frac{1}{\sigma_b} - \frac{1}{\sigma_t} \right) \frac{I_1}{3} \right)}$$

## THE 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

$$g(\sigma) = \frac{aJ_2}{\sigma_c(\sigma_c - bI_1)} + \frac{\lambda(\theta)\sqrt{J_2}}{\sigma_c - bI_1} - 1$$

In this formulation, the parameters  $a$  and  $b$  are positive and dimensionless, and  $\sigma_c$  is the uniaxial compressive strength for concrete (also with a positive sign). The dimensionless function  $\lambda(\theta)$  depends on the Lode angle  $\theta$  and two positive parameters  $k_1$  and  $k_2$ , see [The Ottosen Criterion](#) for details. The failure index is computed from

$$fi = \frac{aJ_2}{\sigma_c(\sigma_c - bI_1)} + \frac{\lambda(\theta)\sqrt{J_2}}{\sigma_c - bI_1}$$

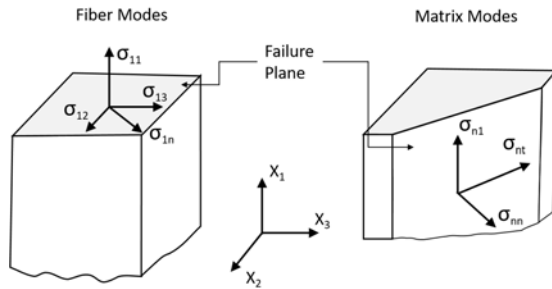
## USER DEFINED

This option allows you to write explicitly how the failure criterion and the safety factor depend on stress and/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.

You can add any number of **Safety** nodes to a single material model. The contents of this feature will not affect the analysis results as such, as this feature does not account for post-failure analysis. 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.

## FIBER COMPOSITE SPECIFIC FAILURE CRITERIA

Apart from Tsai–Wu family criteria, there are certain advanced composite specific criteria. These criteria account for multi-axial state of stress, different failure modes, and different criterion for tensile and compressive loading. For example, in fiber composites, failure can happen because of fiber rupture in tension or fiber buckling in compression, matrix failure in tension or matrix failure in compression.

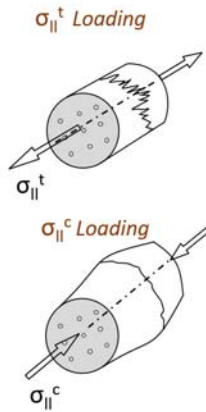


*Figure 3-28: The fiber and matrix failure modes and respective failure planes in a unidirectional composite.*

The basic types of failure modes, under various loadings, in unidirectional composites are as follows (Figure 3-29)

- 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

### Fiber Failure (FF)



### Inter Fiber Failure (IFF)

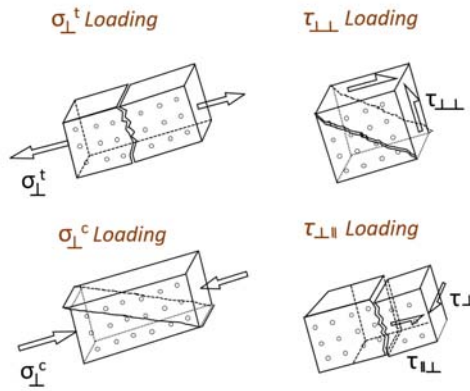


Figure 3-29: Different fiber failure (FF) modes and interfiber failure (IFF) modes of a unidirectional composite under different loading conditions. The plane where the brittle fracture occurs is also shown.

The list of fiber composite specific failure criteria includes:

- Zinoviev Criterion
- The Hashin–Rotem Criterion
- The Hashin Criterion
- Puck Criterion
- LARC-03 Criterion

Table 3-7 shows failure criterion and considered failure modes by that criterion. .

TABLE 3-7: 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 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 is 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 multi-directional laminates (Ref. 19). The approach is one of the most used due to its simplicity, and it is recommended by World Wide Failure Exercise. The failure criteria for different failure modes are 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

$$g_s(\sigma) = \left| \frac{\sigma_{12}}{\sigma_{ss12}} \right| - 1$$

By picking up the most critical failure mode, the failure criterion for composite is written as

$$g(\sigma) = \max(g_{1T}, g_{1C}, g_{tT}, g_{tC}, g_s)$$

### THE HASHIN-ROTEM CRITERION

This criterion is based on the first ply failure theory and designed for unidirectional laminates. The Hashin–Rotem failure theory considers effects of tensile and compressive stresses while predicting the failure of fiber or matrix, however it does not

distinguish whether the failure is exactly at the interface or inside the matrix. This 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

$$g_{fT}(\sigma) = \frac{\sigma_{11}}{\sigma_{ts1}} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criterion in compression is

$$g_{fC}(\sigma) = \left| \frac{\sigma_{11}}{\sigma_{cs1}} \right| - 1 \text{ for } \sigma_{11} < 0$$

The matrix failure criterion in tension is

$$g_{mT}(\sigma) = \left( \frac{\sigma_{22}}{\sigma_{ts2}} \right)^2 + \left( \frac{\sigma_{12}}{\sigma_{ss12}} \right)^2 - 1 \text{ for } \sigma_{22} \geq 0$$

The matrix failure criterion in compression is

$$g_{mC}(\sigma) = \left( \frac{\sigma_{22}}{\sigma_{cs2}} \right)^2 + \left( \frac{\sigma_{12}}{\sigma_{ss12}} \right)^2 - 1 \text{ for } \sigma_{22} < 0$$

By picking up the most critical failure mode, the failure criterion for composite is written as

$$g(\sigma) = \max(g_{fT}, g_{fC}, g_{mT}, g_{mC})$$

### THE HASHIN CRITERION

The Hashin failure theory is extension of Hashin–Rotem failure theory with six failure modes including fiber failure, matrix failure and interlaminar failure, distinguishing between tension and compression (Ref. 18). Stress interactions are considered for the determination of the tensile fiber failure mode, tensile matrix failure mode and compressive matrix failure mode.

The fiber failure criterion in tension is

$$g_{fT}(\sigma) = \frac{\sigma_{11}^2}{\sigma_{ts1}^2} + \frac{\sigma_{12}^2 + \sigma_{13}^2}{\sigma_{ss12}^2} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criterion in compression is

$$g_{fC}(\sigma) = \frac{\sigma_{11}^2}{\sigma_{cs1}^2} - 1 \text{ for } \sigma_{11} < 0$$

The matrix failure criterion in tension is

$$g_{mT}(\sigma) = \frac{(\sigma_{22} + \sigma_{33})^2}{\sigma_{ts2}^2} + \frac{(\sigma_{23}^2 - \sigma_{22}\sigma_{33})}{\sigma_{ss23}^2} + \frac{(\sigma_{12}^2 + \sigma_{13}^2)}{\sigma_{ss12}^2} - 1 \text{ for } \sigma_{22} + \sigma_{33} \geq 0$$

The matrix failure criterion in compression is

$$g_{mC}(\sigma) = \left( \frac{\sigma_{cs2}^2}{4\sigma_{ss23}^2} - 1 \right) \frac{(\sigma_{22} + \sigma_{33})}{\sigma_{cs2}} + \frac{(\sigma_{22} + \sigma_{33})^2}{4\sigma_{ss23}^2} + \frac{(\sigma_{23}^2 - \sigma_{22}\sigma_{33})}{\sigma_{ss23}^2} + \frac{(\sigma_{12}^2 + \sigma_{13}^2)}{\sigma_{ss12}^2} - 1$$

for  $\sigma_{22} + \sigma_{33} < 0$

The interlaminar failure criterion in tension is

$$g_{iT}(\sigma) = \frac{\sigma_{33}^2}{\sigma_{ts3}^2} - 1 \text{ for } \sigma_{33} \geq 0$$

The interlaminar failure criterion in compression is

$$g_{iC}(\sigma) = \frac{\sigma_{33}^2}{\sigma_{cs3}^2} - 1 \text{ for } \sigma_{33} < 0$$

By picking up the most critical failure mode, the failure criterion for composite is written as

$$g(\sigma) = \max(g_{fT}, g_{fC}, g_{mT}, g_{mC}, g_{iT}, g_{iC})$$

For plane stress conditions, the Hashin criterion is modified by setting  $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$ . With 2D plane stress version of Hashin criterion, the interlaminar failure cannot be predicted.

### PUCK CRITERION

The Puck criterion is based on 3-D 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 C correspond to compressive and shear loading.

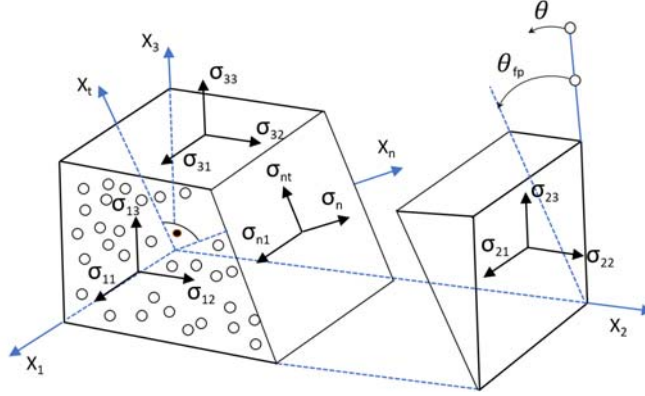


Figure 3-30: Interfiber failure (IFF) mode using Puck criterion.

The Puck criterion is mainly used for predicting strength of unidirectional laminate and for predicting the initial strength of multidirectional laminates, for which the other methods do not predict the failure correctly (Ref. 19). This criterion is also recommended by World Wide Failure Exercise. It is distinguishing and treating separately failure criteria the fiber failure (FF) and interfiber failure (IFF).

The fiber failure criterion in tension is

$$g_{fT}(\sigma, \varepsilon) = \frac{1}{\varepsilon_{ts1}} \left( \varepsilon_{11} + \frac{\nu_{f12}}{E_{f1}} m_{\sigma f} \sigma_{22} \right) - 1 \text{ for } \varepsilon_{11} \geq 0$$

where  $E_{f1}$  and  $\nu_{f12}$  are Young's modulus of fiber in longitudinal direction and in-plane Poisson's ratio of fiber, respectively, and  $m_{\sigma f}$  is the mean stress magnification factor.

The fiber failure criterion in compression is

$$g_{fC}(\sigma, \varepsilon) = \frac{1}{\varepsilon_{cs1}} \left| \left( \varepsilon_{11} + \frac{\nu_{f12}}{E_{f1}} m_{\sigma f} \sigma_{22} \right) \right| + (10\nu_{12})^2 - 1 \text{ for } \varepsilon_{11} < 0$$

The failure criterion for interfiber failure mode A is

$$g_{mA}(\sigma) = \left( \left( \frac{\sigma_{12}}{\sigma_{ss12}} \right)^2 + \left( 1 - \frac{\sigma_{ts2} p_{tl}}{\sigma_{ss12}} \right)^2 \left( \frac{\sigma_{22}}{\sigma_{ts2}} \right)^2 \right)^{\frac{1}{2}} + \frac{\sigma_{22} p_{tl}}{\sigma_{ss12}} + \left| \frac{\sigma_{11}}{\sigma_{1D}} \right| - 1 \text{ for}$$

$$\sigma_{22} \geq 0$$

where  $p_{tl}$  is slope of the in-plane fracture envelope in tension and  $\sigma_{1D}$  is linear degradation stress.

The failure criterion for interfiber failure mode B is

$$g_{mB}(\sigma) = \frac{1}{\sigma_{ss12}} \left( (\sigma_{12}^2 + (\sigma_{22} p_{cl})^2)^{\frac{1}{2}} + \sigma_{22} p_{cl} \right) + \left| \frac{\sigma_{11}}{\sigma_{1D}} \right| - 1 \text{ for } \sigma_{22} < 0 \text{ and}$$

$$0 \leq \left| \frac{\sigma_{22}}{\sigma_{12}} \right| \leq \left| \frac{R_{tt}^A}{\sigma_{ss12}^c} \right|$$

where  $p_{cl}$  is slope of the in-plane fracture envelope in compression,  $R_{tt}^A$  is fracture resistance against transverse shear loading and  $\sigma_{ss12}^c$  is modified shear strength.

The failure criterion for interfiber failure mode C is

$$g_{mC}(\sigma) = \frac{\sigma_{cs2}}{-\sigma_{22}} \left( \left( \frac{\sigma_{12}}{2(1+p_{ct})\sigma_{ss12}} \right)^2 + \left( \frac{\sigma_{22}}{\sigma_{cs2}} \right)^2 \right) + \left| \frac{\sigma_{11}}{\sigma_{1D}} \right| - 1 \text{ for } \sigma_{22} < 0 \text{ and}$$

$$0 \leq \left| \frac{\sigma_{12}}{\sigma_{22}} \right| \leq \left| \frac{\sigma_{ss12}^c}{R_{tt}^A} \right|$$

where  $p_{ct}$  is slope of the transverse fracture envelope in compression.

By picking up, the most critical failure mode, the failure criterion for composite is written as

$$g(\sigma) = \max(g_{fT}, g_{fC}, g_{mA}, g_{mB}, g_{mC})$$

### LARC-03 CRITERION

This criterion is used for accurate predicting the failure of unidirectional FRP laminates with in-plane stress state. This is one of the most advanced failure criteria composed of six phenomenological failure modes describing matrix and fiber failure accurately without the use of curve-fitting parameters (Ref. 20). This criterion assumes a fragile fracture for the matrix failure in compression and implements the action plane concept

according to the Mohr–Coulomb theory. This failure theory considers failure modes based on the fiber kinking due to misalignment and on the tensile matrix cracking associated with interlaminar crack propagation.

The matrix failure criterion under transverse compression (LaRC03-1) is

$$g_{mC}(\sigma) = \left( \frac{\tau_{\text{eff},t}}{\sigma_{ss23}^i} \right)^2 + \left( \frac{\tau_{\text{eff},l}}{\sigma_{ss12}^i} \right)^2 - 1 \text{ for } \sigma_{22} < 0$$

where  $\tau_{\text{eff},t}$  and  $\tau_{\text{eff},l}$  are effective shear stresses in transverse and longitudinal directions, respectively, and  $\sigma_{ss12}^i$  is longitudinal in situ shear strength. The effective shear stresses are functions of the fracture plane angle which is found out by maximizing the Mohr–Coulomb effective stresses.

The matrix failure criterion under transverse tension (LaRC03-2) is

$$g_{mT}(\sigma) = (1-r) \left( \frac{\sigma_{22}}{\sigma_{ts2}^i} \right) + r \left( \frac{\sigma_{22}}{\sigma_{ts2}^i} \right)^2 + \left( \frac{\sigma_{12}}{\sigma_{ss12}^i} \right)^2 - 1 \text{ for } \sigma_{22} \geq 0$$

where  $\sigma_{ts2}^i$  is in situ tensile strength, and  $r$  is a material constant based on fracture toughness.

The fiber failure criterion under longitudinal compression (LaRC03-3) is

$$g_{fT}(\epsilon) = \frac{\epsilon_{11}}{\epsilon_{ts1}} - 1 \text{ for } \sigma_{11} \geq 0$$

The fiber failure criteria with matrix tension (LaRC03-4) is

$$g_{fCmT}(\sigma) = (1-r) \left( \frac{\sigma_{22}^m}{\sigma_{ts2}^i} \right) + r \left( \frac{\sigma_{22}^m}{\sigma_{ts2}^i} \right)^2 + \left( \frac{\sigma_{12}^m}{\sigma_{ss12}^i} \right)^2 - 1 \text{ for } \sigma_{22}^m \geq 0$$

The fiber failure criteria with matrix compression (LaRC03-5) is

$$g_{fCmC}(\sigma) = \left\langle \frac{|\sigma_{12}^m| + \sigma_{22}^m \eta_1}{\sigma_{ss12}^i} \right\rangle - 1 \text{ for } \sigma_{22}^m < 0$$

where  $\sigma_{ij}^m$  are the ply stresses transformed in the misalignment coordinate frame, and  $\eta_1$  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

$$g_{mB}(\sigma) = \left( \frac{\tau_{ss23}^m}{\sigma_{ss23}} \right)^2 + \left( \frac{\tau_{ss12}^m}{\sigma_{ss12}^i} \right)^2 - 1 \text{ for } \sigma_{11} < 0 \text{ and } \sigma_{22} < 0$$

where the effective shear stresses in transverse and longitudinal directions,  $\tau_{\text{eff},t}^m$  and  $\tau_{\text{eff},l}^m$ , are calculated from stresses in the misalignment coordinate frame.

By picking up the most critical failure mode, the failure criterion for composite is written as

$$g(\sigma) = \max(g_{mC}, g_{mT}, g_{fT}, g_{fCmT}, g_{fCmC}, g_{mB})$$

# 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  $\boldsymbol{\sigma}$ :

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_x \boldsymbol{\sigma} + \mathbf{f}_V$$

Here  $\mathbf{f}_V$  is a body force per unit deformed volume, and  $\rho$  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:

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_X \mathbf{P}^T + \mathbf{F}_V$$

Now that the first Piola–Kirchhoff stress tensor,  $\mathbf{P}$ , is used.  $\mathbf{F}_V$  is a body force with components in the current configuration but given with respect to the undeformed volume, and  $\rho_0$  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,  $\mathbf{S}$ , the same equation reads

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_X (\mathbf{F}\mathbf{S}) + \mathbf{F}_V \quad (3-124)$$

where  $\mathbf{F}$  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  $\delta W$  is written as

$$\delta W = \int_V (-\delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} + \delta\mathbf{u} \cdot \mathbf{F}_V) dv + \int_S (\delta\mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta\mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta\mathbf{u} \cdot \mathbf{F}_p)$$

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.

$$\delta W = \int_V (-\delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} + \delta\mathbf{u} \cdot \mathbf{F}_V - \rho \delta\mathbf{u} \cdot \mathbf{u}_{tt}) dv + \int_S (\delta\mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta\mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta\mathbf{u} \cdot \mathbf{F}_p) \quad (3-125)$$

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  $\boldsymbol{\varepsilon}$  is the engineering strain. The stress  $\boldsymbol{\sigma}$  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  $\delta\boldsymbol{\varepsilon}$  is taken from the displacement gradient, and the stress conjugate  $\boldsymbol{\sigma}$  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,  $\phi_m$

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

$$u = \text{Re}(u_{\text{amp}} e^{j\phi_u} e^{j\omega t}) = \text{Re}(\tilde{u} e^{j\omega t}) \quad \text{where } \tilde{u} = u_{\text{amp}} e^{j\phi_u}$$

$$\mathbf{u} = \text{Re}(\tilde{\mathbf{u}}e^{j\omega t})$$

and

$$F_{m, \text{freq}} = \text{Re}(F_m(\omega)e^{j\phi_m}e^{j\omega t}) = \text{Re}(\tilde{F}_m e^{j\omega t})$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{bmatrix}$$

The primary results, such as displacements, velocities, accelerations, and stress and strain components are all complex valued.

### *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  $\lambda$  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,  $\delta$ , for the model can be examined:

$$Q = \frac{\text{Im}(\lambda)}{2\text{Re}(\lambda)}$$

$$\delta = \text{Re}(\lambda)$$

### **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  $i$ -th 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  $\phi_i$  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  $\psi_j^m$  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  $\mathbf{u} = [u \ v \ w]$ . The rigid body modes  $\psi$  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:

$$\Gamma_T = [\Gamma_{TX} \ \Gamma_{TY} \ \Gamma_{TZ}] = \frac{L}{m_F} \int \mathbf{u} dm$$

and

$$\Gamma_R = [\Gamma_{RX} \ \Gamma_{RY} \ \Gamma_{RZ}] = \frac{1}{m_F} \int (\mathbf{r} - \mathbf{r}_0) \times \mathbf{u} dm$$

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** etc. 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  $\theta = [\theta_X \ \theta_Y \ \theta_Z]$  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

$$m_{TX}^{\text{eff}} = \frac{m_F}{L^2} \Gamma_{TX}^2 = M(\Gamma_{TX}^{\text{norm}})^2$$

and

$$m_{RX}^{\text{eff}} = m_{FX} \Gamma_R^2$$

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 of 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

$$E[x] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t) dt = m$$

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,  $x_{\text{rms}}$ .

A representation of the ‘degree of similarity over time’ of a signal is given by the autocorrelation function

$$R_x(\tau) = E[x(t)x(t + \tau)] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t)x(t + \tau) dt$$

It can be seen from the definition that the autocorrelation is an even function of the time difference  $\tau$ . 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

$$R_{xy}(\tau) = E[x(t)y(t + \tau)] = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t)y(t + \tau) dt$$

The power spectral density (PSD) of a signal  $x$  is defined through the Fourier transform (denoted by  $\mathbf{F}[\cdot]$ ) of the autocorrelation,

$$G_x(f) = 2\mathbf{F}[R_x(\tau)] = 2 \int_{-\infty}^{\infty} R_x(\tau) e^{-2\pi i f \tau} d\tau$$

The term PSD comes from the fact that the physical dimension of  $G_x$  is (input signal)<sup>2</sup> / frequency. Thus,  $G_x(f)\Delta f$  represents the power of the signal contained in the small frequency interval  $\Delta f$ .

The inverse relation is

$$R_x(\tau) = \frac{1}{2} \mathcal{F}^{-1}[G_x(f)] = \frac{1}{2} \int_{-\infty}^{\infty} G_x(f) e^{2\pi i f \tau} df$$

Since the autocorrelation is a real-valued and even function of  $\tau$ ,  $G_x$  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.

$$x_{\text{rms}}^2 = R_x(0) = \frac{1}{2} \int_{-\infty}^{\infty} G_x(f) df = \int_0^{\infty} G_x(f) df$$

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

$$G_{xy}(f) = 2\mathcal{F}[R_{xy}(\tau)]$$

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

$$x_{rms}^2 = \int_0^{\infty} S_x(\Omega) d\Omega$$

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

$$x_{rms} = \sqrt{m_0}$$

- Average number of (upward) zero crossings per time unit, also called *apparent frequency*

$$n_0 = \sqrt{\frac{m_2}{m_0}}$$

- Average number of peaks per time unit

$$n_p = \sqrt{\frac{m_4}{m_2}}$$

- Irregularity factor

$$\gamma = \frac{n_0}{n_p}$$

### 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

$$S_u(\Omega) = |H_{ux}(\Omega)|^2 S_x(\Omega) = H_{ux}^*(\Omega)H_{ux}(\Omega)S_x(\Omega)$$

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

$$S_u = \sum_k |H_{ux_k}(\Omega)|^2 S_{x_k}$$

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  $M \times N$ , 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 nonhomogeneous 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 gravity load.

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

$$\mathbf{m}\ddot{\mathbf{q}} + \mathbf{c}\dot{\mathbf{q}} + \mathbf{k}\mathbf{q} = \mathbf{l}$$

Lower case matrices are used to indicate a quantity in the modal space.

The projections for the matrices are

$$\begin{aligned}\mathbf{m} &= \mathbf{Y}^T \mathbf{M} \mathbf{Y} \\ \mathbf{c} &= \mathbf{Y}^T \mathbf{C} \mathbf{Y} \\ \mathbf{k} &= \mathbf{Y}^T \mathbf{K} \mathbf{Y} \\ \mathbf{l} &= \mathbf{Y}^T \mathbf{L}\end{aligned}$$

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

$$\begin{aligned} y(t) &= \mathbf{a}^T \mathbf{u}(t) \\ z(t) &= \mathbf{b}^T \mathbf{u}(t) \end{aligned}$$

Then, by using the definitions above and the linearity of the Fourier transform,

$$S_{yz}(\Omega) = \frac{1}{\pi} \mathbf{F}[R_{xy}(\tau)] = \frac{1}{\pi} \mathbf{F}[\mathbf{a}^T \mathbf{R}_{\mathbf{u}\mathbf{u}}(\tau) \mathbf{b}] = \mathbf{a}^T \mathbf{S}_U(\Omega) \mathbf{b} = \mathbf{a}^T \mathbf{Y}\mathbf{s}_q(\Omega)\mathbf{Y}^T \mathbf{b}$$

### *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:

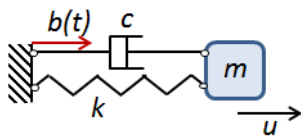
- [Single Degree of Freedom System](#)
- [Definition of a Response Spectrum](#)
- [Solution Using Response Spectrum](#)

- 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,

$$\ddot{u} + 2\zeta\omega_0\dot{u} + \omega_0^2u = 2\zeta\omega_0\dot{b} + \omega_0^2b$$

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  $\omega_0$  and  $\zeta$ , 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

$$\ddot{u}_r + 2\zeta\omega_0\dot{u}_r + \omega_0^2u_r = -\ddot{b} \quad (3-126)$$

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  $\omega_0$ ,  $\zeta$ , and  $b(t)$ , it is a trivial task to solve [Equation 3-126](#) 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

$$\begin{aligned}
S_a(\omega_0, \zeta, b(t)) &= \max |\ddot{u}(t, \omega_0, \zeta)| \\
S_v(\omega_0, \zeta, b(t)) &= \max |\dot{u}(t, \omega_0, \zeta)| \\
S_d(\omega_0, \zeta, b(t)) &= \max |u(t, \omega_0, \zeta)|
\end{aligned}$$

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

$$\begin{aligned}
S_v &\approx \omega_0 S_d \\
S_a &\approx \omega_0^2 S_d
\end{aligned}$$

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**

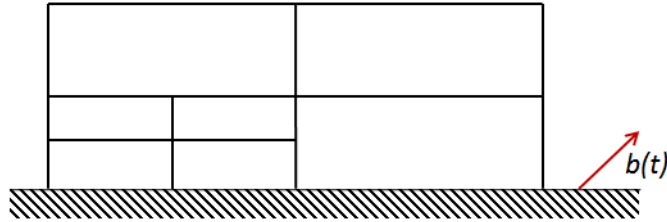
Assume that one has a structure discretized by FEM, so that the equations of motion on matrix form are

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t) \tag{3-127}$$

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 3-127 can be written as

$$\mathbf{M}(\ddot{\mathbf{u}}_r + \dot{\mathbf{b}}) + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = \mathbf{0}$$

or

$$\mathbf{M}\ddot{\mathbf{u}}_r + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = -\mathbf{M}\dot{\mathbf{b}}$$

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

$$(\mathbf{K} - \omega^2\mathbf{M})\boldsymbol{\phi} = \mathbf{0}$$

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

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2q_j = -\phi_j^T \mathbf{M}\ddot{\mathbf{b}} \quad (3-128)$$

The equation is similar to the standard SDOF system in [Equation 3-126](#).



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 3-128](#),  $q_j$  is the modal coordinate for mode  $j$ , so that the relative displacement can be written as a superposition of the eigenmodes  $\phi_j$ .

$$\mathbf{u}_r = \sum_j \phi_j q_j = \Phi \mathbf{q}$$

The support motion can be written as

$$\mathbf{b}(t) = b_x(t)\mathbf{1}_x + b_y(t)\mathbf{1}_y + b_z(t)\mathbf{1}_z \quad (3-129)$$

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 3-129](#) in [Equation 3-128](#) gives

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2q_j = -\Gamma_{xj}\ddot{b}_x(t) - \Gamma_{yj}\ddot{b}_y(t) - \Gamma_{zj}\ddot{b}_z(t)$$

The multipliers  $\Gamma_{kj}$  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

$$\hat{q}_{kj} = S_d(\omega_j, \zeta_j, b_k(t)) \Gamma_{kj} = S_{d,k} \Gamma_{kj}$$

or, using the pseudoacceleration spectrum

$$\hat{q}_{kj} = \frac{S_a(\omega_j, \zeta_j, b_k(t)) \Gamma_{kj}}{\omega_j^2} = \frac{S_{a,k} \Gamma_{kj}}{\omega_j^2}$$

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  $R_{pI}$ , and the rigid part is denoted  $R_{rI}$ . Similarly,  $R_{pIj}$  and  $R_{rIj}$  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.

### Partitioning into Periodic and Rigid Modes

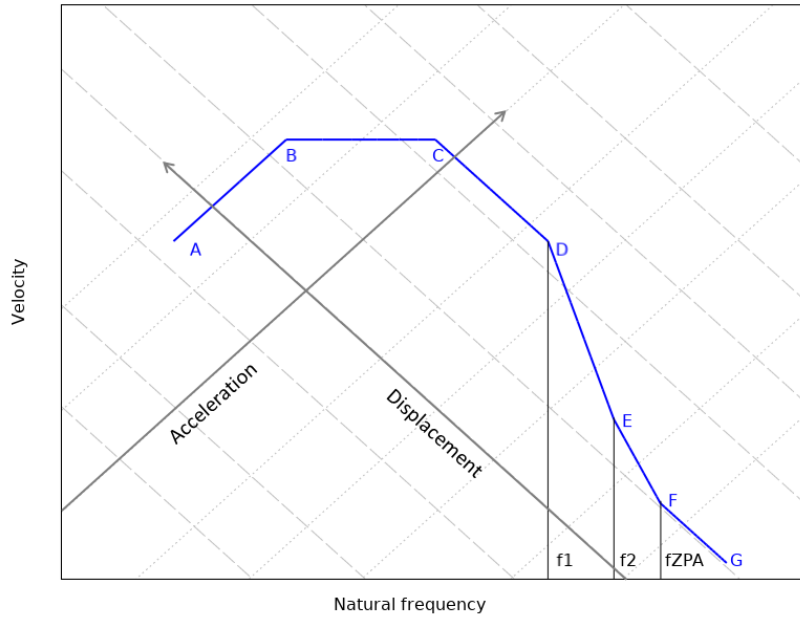


Figure 3-31: A schematic tripartite plot of a design response spectrum. Both axes have logarithmic scales.

There are two different methods in use, by which the partitioning can be done. In either case, for mode  $j$ ,

$$R_{rI,j} = \alpha_j R_{I,j}$$

$$R_{pI,j} = \sqrt{1 - \alpha_j^2} R_{I,j}$$

This definition has the property that

$$R_{I,j}^2 = R_{pI,j}^2 + R_{rI,j}^2$$

The difference between the two methods lies in how the coefficients  $\alpha_j$  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,  $\alpha_j$  is a linear function of the logarithm of the natural frequency.

$$\begin{aligned} \alpha_j &= 0 & f_j &\leq f_1 \\ \alpha_j &= \frac{\log(f_j/f_1)}{\log(f_2/f_1)} & f_1 &< f_j < f_2 \\ \alpha_j &= 1 & f_j &\geq f_2 \end{aligned}$$

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

$$f_1 = \frac{S_{a,\max}}{2\pi S_{v,\max}}$$

where  $S_{a,\max}$  and  $S_{v,\max}$  are the maximum values of the acceleration and velocity spectra, respectively. In the idealized spectrum shown in [Figure 3-31](#), 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  $\alpha_j$  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

$$R_{pI,j} = \frac{\sqrt{S_{a,I}^2(f_j) - S_{ZPA,I}^2}}{S_{a,I}(f_j)} R_{I,j}$$

The so-called *zero period acceleration* (ZPA) is the maximum ground acceleration during the event

$$S_{ZPA,I} = \max_t |\ddot{b}_I(t)|$$

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 3-31](#).

Thus,

$$\alpha_{j,I} = \frac{S_{ZPA,I}}{S_{a,I}(f_j)}$$

The value of  $\alpha_j$  must be in the range from 0 to 1, and it must increase with frequency. For this reason, NRC RG 1.92 requires that  $\alpha_j$  must be set to zero for any eigenmodes below point C in [Figure 3-31](#). For a general spectrum, this is implemented as a strict requirement that  $\alpha_j$  has a monotonous decrease with decreasing frequency from  $f_{ZPA}$ . As soon as an increase in  $\alpha_j$  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

$$R_I = R_{pI}^2 + R_{rI}^2$$

#### *Summing the Periodic Modes*

All summation rules for periodic modes except [Absolute Value Sum](#) can be summarized by the following expression:

$$R_{pI}^2 = \sum_{i=1}^N \sum_{j=1}^N C_{ij} R_{pI,i} R_{pI,j}$$

Here,  $R_{pI}$  is the total periodic response of some result quantity  $R$  with respect to excitation in direction  $I$  ( $I=1,2,3$ ).  $R_{pI,j}$  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  $C_{ij}$  ( $0 \leq C_{ij} \leq 1$ ). 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

$$R_{pI}^2 = \sum_{i=1}^N R_{pI,i}^2 + 2 \sum_{i=1}^N \sum_{j=i+1}^N C_{ij} R_{pI,i} R_{pI,j}$$

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)

$$R_{I,j} = g(S_{d,I} \Gamma_{Ij} \phi_j)$$

Next, the different evaluation methods are presented.

#### *SRSS Method*

In the SRSS method, it is assumed that the modes are statistically independent, so that

$$\begin{aligned} C_{ij} &= 1 & i = j \\ C_{ij} &= 0 & i \neq j \end{aligned}$$

#### *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  $f_i \leq 1.1f_k$ , 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

$$\begin{aligned} C_{ij} &= \text{sgn}(R_{pI,i}R_{pI,j}) && \text{when the modes are in the same group} \\ C_{ij} &= 0 && \text{when the modes are not in the same group} \end{aligned}$$

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

$$\begin{aligned} C_{ij} &= 1 && \text{when the modes are in the same group} \\ C_{ij} &= 0 && \text{when the modes are not in the same group} \end{aligned}$$

#### *Ten Percent Method*

The ten percent method is similar to the grouping method, and has the following definition:

$$\begin{aligned} C_{ij} &= \text{sgn}(R_{pI,i}R_{pI,j}) && f_i \leq 1.1f_j \\ C_{ij} &= 0 && \text{else} \end{aligned}$$

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

$$\begin{aligned} C_{ij} &= 1 & f_i \leq 1.1f_j \\ C_{ij} &= 0 & \text{else} \end{aligned}$$

#### 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

$$C_{ij} = \text{sgn}(R_{pI,i}R_{pI,j}) \left[ 1 + \left( \frac{f_i' - f_j'}{\zeta_i' f_i + \zeta_j' f_j} \right)^2 \right]^{-1}$$

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  $\zeta_i$  is the modal damping. In the implementation in COMSOL Multiphysics, all modes are assumed to have the same damping.

The modified damping,  $\zeta_i'$ , is defined as

$$\zeta_i' = \zeta_i + \frac{1}{\pi t_d f_i}$$

where  $t_d$  is a separate input, called the time of duration. The value of  $\zeta_i$  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

$$C_{ij} = \frac{8\zeta^2 \sqrt{f_i f_j} (f_i + f_j) f_i f_j}{(f_i^2 - f_j^2)^2 + 4\zeta^2 f_i f_j (f_i + f_j)^2} \quad (3-130)$$

#### 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.

#### 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

$$R_{rI} = \sum_{i=1}^N R_{rI,i} + R_{mm,I}$$

where  $R_{mm,I}$  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

$$R_{rI} = R_{\text{StaticZPA},I}$$

#### 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 devise 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  $\mathbf{f}_c$  can be written as

$$\mathbf{f}_c = \mathbf{f} - \sum_k r_k (\mathbf{M}\phi_k)$$

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  $\phi$ ,

$$r_k = \phi_k^T \mathbf{f}$$

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

$$\mathbf{f}_{c,I} = -\mathbf{M}\ddot{b}_I\mathbf{1}_I + \sum_k \Gamma_{Ik}\ddot{b}_I(\mathbf{M}\phi_k)$$

For the rigid body modes, the maximum ground acceleration during the event is equal to the ZPA. The static load is thus

$$\mathbf{f}_{c,I} = S_{ZPA}\mathbf{M}\left(-\mathbf{1}_I + \sum_k \Gamma_{Ik}\phi_k\right) \quad (3-131)$$

The extra displacement correcting for the missing mass is then given by the standard stationary problem

$$\mathbf{K}\mathbf{u}_{rc,I} = \mathbf{f}_{c,I}$$

To actually compute the load in [Equation 3-131](#), 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

$$\sum_k \Gamma_{Ik} \phi_k - \mathbf{1}_I \quad (3-132)$$

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

$$R_{pI}^2 = \sum_{i=1}^N \sum_{j=1}^N C_{ij} R_{pI,i} R_{pI,j}$$

where the Der Kiureghian expression [Equation 3-130](#) 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  $\theta$  with respect to the  $X-Y$  orientations. It can then be shown that

$$\begin{aligned}R_{pX,i} &= R_{pX,i} \cos \theta + R_{pY,i} \sin \theta \\R_{pY,i} &= R_{pY,i} \cos \theta - R_{pX,i} \sin \theta\end{aligned}$$

Also, if the relation between the two spectra in the horizontal plane is such that

$$S_Y = \gamma S_X$$

the same ratio  $\gamma$  will apply to the responses. The peak response as function of the rotation angle is obtained by an SRSS type summation

$$\begin{aligned}R(\theta) &= \{ [R_{pX}^2 + \gamma^2 R_{pY}^2] \cos^2 \theta + [R_{pY}^2 + \gamma^2 R_{pX}^2] \sin^2 \theta + \\ &\quad 2 \sin \theta \cos \theta (1 - \gamma^2) R_{pXY}^2 + R_{pZ}^2 \}^{1/2}\end{aligned}$$

It can be seen that for  $\gamma = 1$ , the standard SRSS expression is retrieved.

The angle  $\theta_{\max}$  giving the maximum response  $R(\theta_{\max})$  turns out to be independent of  $\gamma$ , 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  $\theta_{\max}$ , 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:

$$R^2 = \sum_{I=1}^3 R_I^2 = \sum_{I=1}^3 (R_{pI}^2 + R_{rI}^2) = \sum_{I=1}^3 \left( \sum_{i=1}^N \sum_{j=1}^N C_{ij} R_{pI,i} R_{pI,j} + R_{rI}^2 \right)$$

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

$$R_{rXY}^2 = R_{rX}R_{rY}$$

Now, it is possible to have the rigid modes too 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

$$\begin{cases} C_{N+1N+1} = 1 \\ C_{iN+1} = 0 \quad \text{when } i \neq N+1 \end{cases}$$

In the GUI, this extension is selected by checking the **Augment with rigid response** check box.

#### *SRSS3 Method*

The SRSS3 method is a special case of the CQC3 rule, in which the mode correlation is ignored, that is

$$\begin{aligned} C_{ij} &= 1 & i &= j \\ C_{ij} &= 0 & i &\neq j \end{aligned}$$

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.



For information about how to perform a response spectrum analysis, see [Response Spectrum Analysis Study](#)

---

#### *Linear Buckling*

The 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  $\lambda$  is the multiplier to the original load that would cause buckling.



For a general introduction, see [Linearized Buckling Analysis](#)

---

The formulation in terms of virtual work is

$$\delta W = \int_V (-\delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} - \lambda \delta(\boldsymbol{\varepsilon}_{\text{GL}} - \boldsymbol{\varepsilon}) : \boldsymbol{\sigma}_1) dv = 0 \quad (3-133)$$

Here  $\boldsymbol{\varepsilon}$  is the engineering strain,  $\boldsymbol{\varepsilon}_{\text{GL}}$  is the Green–Lagrange strain and  $\boldsymbol{\sigma}_1$  is the stress caused by the unit load. In terms of stiffness matrices, this corresponds to

$$(\mathbf{K}_L + \lambda \mathbf{K}_{\text{NL}}(\mathbf{u}_0)) \mathbf{u} = 0$$

where  $\mathbf{K}_L$  is the linear stiffness matrix, and  $\mathbf{K}_{\text{NL}}$  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 3-133, while the first term uses the linear (engineering) strain tensor.

If, however, geometric nonlinearity is selected in the linear buckling study step, Equation 3-133 is replaced by

$$\delta W = \int_V (-\delta \boldsymbol{\varepsilon}_{\text{GL}} : \boldsymbol{\sigma} - (\lambda - 1) \delta(\boldsymbol{\varepsilon}_{\text{GL}} - \boldsymbol{\varepsilon}) : \boldsymbol{\sigma}_1) dv = 0$$

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 Theory](#)
- [Loss Factor Damping Theory](#)
- [Viscous Damping Theory](#)

## *Rayleigh Damping Theory*

---

Rayleigh damping is described by two coefficients: the mass damping coefficient  $\alpha_{dM}$  and the stiffness damping coefficient  $\beta_{dK}$ . 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. If you enter the same Rayleigh damping parameters for all materials, and solve a pure structural mechanics problem, then the classical definition will be retrieved.

### *Loss Factor Damping Theory*

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  $\eta_s$ , 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

$$D_{mn}^c = (1 + j\eta_{s,mn})D_{mn}$$

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:

$$\begin{aligned}
\mathbf{E}_m^c &= (1 + j\eta_{E,m})\mathbf{E}_m \\
\mathbf{G}_m^c &= (1 + j\eta_{G,m})\mathbf{G}_m \\
m &= 1, 2, 3
\end{aligned}$$

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:

$$\mathbf{S}_q = j\eta_s \frac{\partial W_s}{\partial \mathbf{E}}$$

For loss factor damping, the following definition is used for the elastic part of the entropy:

$$\mathbf{S}_{\text{elast}} = \alpha : (s - j\eta_s(\mathbf{C} : \boldsymbol{\varepsilon}))$$

note that  $\mathbf{S}_{\text{elast}}$  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:

$$Q_h = \frac{1}{2}\omega\eta_s \text{real}(\boldsymbol{\varepsilon} : \text{conj}(\mathbf{C} : \boldsymbol{\varepsilon}))$$

$Q_h$  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 Theory*

---

Viscous damping can be added to the material models. It will cause an extra stress  $\mathbf{S}_q$  proportional to the rate of elastic strain in the material,

$$s_q = \eta_b \dot{\boldsymbol{\varepsilon}}_{\text{el, vol}} + \eta_v \dot{\boldsymbol{\varepsilon}}_{\text{el, dev}}$$

where  $\eta_b$  and  $\eta_v$  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

$$\mathbf{S}_q = \left( \eta_b - \frac{2}{3} \eta_v \right) \dot{J}_{\text{el}} \mathbf{C}_{\text{el}}^{-1} + \eta_v \mathbf{C}_{\text{el}}^{-T} \dot{\mathbf{e}}_{\text{el}} \mathbf{C}_{\text{el}}^{-1}$$

where  $\mathbf{C}_{\text{el}}^{-1}$  is the inverse of the elastic Cauchy–Green tensor, and  $J_{\text{el}}$  is the elastic volume ratio.

# Loads and Boundary Conditions

In this section:

- Distributed Loads
- Rotating Frame and Gravity
- Spring Foundation and Thin Elastic Layer
- Added Mass
- Rigid Motion Suppression Boundary Condition
- Low-Reflecting Boundary Condition
- Cyclic Symmetry and Floquet Periodic Conditions
- Symmetry Condition with Translation
- Roller Boundary Condition
- Thermal Expansion of Constraints
- Fluid-Structure Interaction Theory
- Periodic Cell Theory
- Rigid Connector Theory
- Contact Analysis Theory

## *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

$$\mathbf{F} = p \mathbf{n} \frac{da(R+u)}{dA R}$$

---

#### *Rotating Frame and Gravity*

You can add **Rotating Frame** and **Gravity** nodes to create the loads caused by gravity or accelerated frames. This gives load contributions from all nodes in the physics interface which have a density or mass, such as Linear Elastic Material, Rigid Domain, 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 **Gravity** nodes.

---

In the following, the mass density  $\rho$  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.

### 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 acceleration effects:

$$\mathbf{a}_{\text{tot}} = \ddot{\mathbf{u}} + \mathbf{a}_{\mathbf{f}}$$

$$\mathbf{a}_{\mathbf{f}} = \mathbf{a}_{\text{cen}} + \mathbf{a}_{\text{cor}} + \mathbf{a}_{\text{eul}}$$

### GRAVITY

The gravity acts in a fixed spatial direction  $\mathbf{e}_g$ . The intensity is

$$\mathbf{g} = \rho g \mathbf{e}_g$$

where  $g$  is the acceleration of gravity. The action of gravity can also be presented as a linearly accelerated frame of reference. Thus, it can be accounted for as a contribution into the effective acceleration via the frame acceleration term given by:

$$\mathbf{a}_{\mathbf{f}} = -\mathbf{g}$$

### CENTRIFUGAL FORCE

A centrifugal force acts radially outward from the axis of rotation defined by the axial direction vector  $\mathbf{e}_{\text{ax}}$ . The rotation is represented by the angular velocity vector:

$$\boldsymbol{\Omega} = \Omega \mathbf{e}_{\text{ax}}$$

where  $\Omega$  is the angular velocity. In vector form, the acceleration contribution and the loads are:

$$\mathbf{a}_{\text{cen}} = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_p)$$

$$\mathbf{F}_{\text{cen}} = -\rho \mathbf{a}_{\text{cen}}$$

where  $\mathbf{r}_p$  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  $\mathbf{r}_{bp}$ .

### SPIN-SOFTENING EFFECT

The structural displacement can be accounted for when computing the rotation position, so that

$$\mathbf{r}_p = \mathbf{X} + \mathbf{u} - \mathbf{r}_{bp}$$

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:

$$\mathbf{a}_{cor} = 2\Omega \times \frac{\partial \mathbf{u}}{\partial t}$$

$$\mathbf{F}_{cor} = -\rho \mathbf{a}_{cor}$$

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:

$$\mathbf{a}_{eul} = \frac{\partial \Omega}{\partial t} \times \mathbf{r}_p$$

$$\mathbf{F}_{eul} = -\rho \mathbf{a}_{eul}$$

### *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

$$\mathbf{f}_s = -\mathbf{K} \cdot (\mathbf{u} - \mathbf{u}_0)$$

where  $\mathbf{f}_s$  is a force/unit area,  $\mathbf{u}$  is the displacement deforming the spring, and  $\mathbf{K}$  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

$$\mathbf{f}_l = -i\eta\mathbf{K} \cdot (\mathbf{u} - \mathbf{u}_0)$$

where  $\eta$  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  $\mathbf{K}$ .

If the elastic part of the spring definition is given as a force versus displacement relation, the stiffness  $\mathbf{K}$  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,

$$\begin{aligned}\mathbf{f}_{sD} &= -\mathbf{f}_{sS} = \mathbf{f}(\mathbf{u} - \mathbf{u}_0) \\ \mathbf{u} &= \mathbf{u}_D - \mathbf{u}_S\end{aligned}$$

The viscous and structural damping forces have analogous properties,

$$\begin{aligned}\mathbf{f}_{vD} &= -\mathbf{f}_{vS} = -D(\dot{\mathbf{u}}_D - \dot{\mathbf{u}}_S - \dot{\mathbf{u}}_0) \\ \mathbf{f}_{lD} &= -\mathbf{f}_{lS} = -i\eta K(\mathbf{u}_D - \mathbf{u}_S - \mathbf{u}_0)\end{aligned}$$

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. If there is a difference in mesh density on the two sides of the pair, you should select the side with the finer mesh as destination.

### 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

$$\begin{aligned}\mathbf{f}_{su} &= -\mathbf{f}_{sd} = \mathbf{f}(\mathbf{u} - \mathbf{u}_0) \\ \mathbf{u} &= \mathbf{u}_u - \mathbf{u}_d\end{aligned}$$

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>.etelij`, 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  $\mathbf{e}_{aux} = \mathbf{e}_X$ . If the *X* direction cannot be used, the *Y* direction is instead used as the auxiliary direction,  $\mathbf{e}_{aux} = \mathbf{e}_Y$ .
- 2 The local *y* direction is obtained from the part of the auxiliary direction which is orthogonal to the normal direction  $\mathbf{n}$ :

$$\mathbf{e}_y = \frac{\mathbf{e}_{aux} - (\mathbf{e}_{aux} \cdot \mathbf{n})\mathbf{n}}{|\mathbf{e}_{aux} - (\mathbf{e}_{aux} \cdot \mathbf{n})\mathbf{n}|}$$

- 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 3-32](#)), 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.

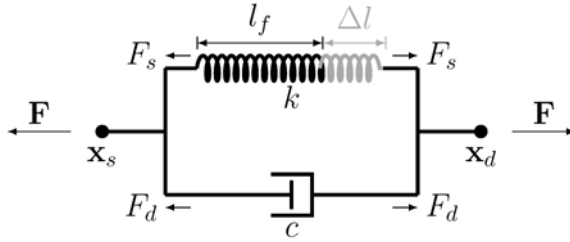


Figure 3-32: Conceptual sketch of a Spring-Damper.

- 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

$$\begin{aligned}\mathbf{x}_s &= \mathbf{X}_s + \mathbf{u}_s \\ \mathbf{x}_d &= \mathbf{X}_d + \mathbf{u}_d\end{aligned}$$

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

$$l = \frac{(\mathbf{x}_d - \mathbf{x}_s) \cdot (\mathbf{X}_d - \mathbf{X}_s)}{l_0}$$

In addition to the initial geometrical distance between the two points you can specify an initial spring extension  $\Delta l_0$ , so that the free length of the spring is

$$l_f = l_0 - \Delta l_0$$

You can also specify the free length of the spring explicitly.

The spring extension  $\Delta l$  is computed as the difference between the current spring length and the free length,

$$\Delta l = l - l_f$$

#### *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  $\Theta_s$  and  $\Theta_d$  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  $\Delta q$  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 check box **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

$$F_s = k(\Delta l) \cdot \Delta l$$

You can also 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>.d1, for example `solid.spd1.d1`.

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

$$\mathbf{F}_{dst} = -\mathbf{F}_{src} = -F \frac{(\mathbf{X}_d - \mathbf{X}_s)}{l}$$

The contribution to the virtual work is

$$\mathbf{F}_{dst} \cdot (\delta \mathbf{u}_d - \delta \mathbf{u}_s)$$

*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

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_s \\ \mathbf{M}_s \end{bmatrix} + \begin{bmatrix} \mathbf{F}_d \\ \mathbf{M}_d \end{bmatrix} = \begin{bmatrix} \mathbf{k}_u & \mathbf{k}_{u\theta} \\ \mathbf{k}_{\theta u} & \mathbf{k}_{\theta} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \Theta \end{bmatrix} + \begin{bmatrix} \mathbf{c}_u & \mathbf{c}_{u\theta} \\ \mathbf{c}_{\theta u} & \mathbf{c}_{\theta} \end{bmatrix} \begin{bmatrix} \Delta \dot{\mathbf{u}} \\ \Delta \dot{\Theta} \end{bmatrix}$$

where the subscripts  $s$  and  $d$  refer to the spring and damper, respectively; the different  $\mathbf{k}$  and  $\mathbf{c}$  represent sub-matrices of size  $3 \times 3$ , 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



$$W_s = i_{ac} \int_0^{\Delta l} F_s d(\Delta l)$$

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

$$W_s = i_{ac} \frac{1}{4} \text{Re}(\text{lindev}(F_s) \cdot \text{conj}(\text{lindev}(\Delta l)))$$

$$W_d = i_{ac} \frac{1}{4} \text{Re}(\text{lindev}(F_d) \cdot \text{conj}(\text{lindev}(\Delta l)))$$

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

$$W_s = i_{ac} \frac{1}{4} [\text{Re}(\text{lindev}(\mathbf{F}_s) \cdot \text{conj}(\text{lindev}(\Delta \mathbf{u}))) +$$

$$\text{Re}(\text{lindev}(\mathbf{M}_s) \cdot \text{conj}(\text{lindev}(\Delta \Theta)))]$$

$$W_d = i_{ac} \frac{1}{4} [\text{Re}(\text{lindev}(\mathbf{F}_d) \cdot \text{conj}(\text{lindev}(\Delta \mathbf{u}))) +$$

$$\text{Re}(\text{lindev}(\mathbf{M}_d) \cdot \text{conj}(\text{lindev}(\Delta \Theta)))]$$

## *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. You can 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

$$\mathbf{f}_m = -M \frac{\partial^2 \mathbf{u}}{\partial t^2}$$

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:

$$\delta W = \int_A \mathbf{f}_m \cdot \delta \mathbf{u} dA$$

## *Rigid Motion Suppression Boundary Condition*

---

The [Rigid Motion Suppression](#) boundary condition is a convenient way to automatically create a set of constraints which are sufficient to inhibit any rigid body modes. The constraints are selected so that no reaction forces are introduced as long as the external loads are in equilibrium.

### **3D WITH TRANSLATIONAL DEGREES OF FREEDOM**

In this case, which includes the Solid Mechanics, Multibody Dynamics, Membrane, and Truss interfaces, six degrees of freedom must be constrained. As it is not possible to directly constrain rotations, this must be done by a proper selection of locations and orientations for the constraints.

The following scheme is used:

- 1 Select three points  $p_1$ ,  $p_2$ , and  $p_3$  that are not located on a common straight line.
- 2 Compute the unit vectors from  $p_1$  to  $p_2$  and  $p_3$ :

$$\mathbf{e}_{12} = \frac{\mathbf{X}_2 - \mathbf{X}_1}{|\mathbf{X}_2 - \mathbf{X}_1|}$$

$$\mathbf{e}_{13} = \frac{\mathbf{X}_3 - \mathbf{X}_1}{|\mathbf{X}_3 - \mathbf{X}_1|}$$

- 3 Compute the normal to the plane spanned by these two vectors.

$$\mathbf{e}_{n1} = \frac{\mathbf{e}_{12} \times \mathbf{e}_{13}}{|\mathbf{e}_{12} \times \mathbf{e}_{13}|}$$

- 4 This normal is perpendicular to the line between  $p_1$  and  $p_2$ . Compute a second perpendicular direction, orthogonal to  $\mathbf{e}_{n1}$ .

$$\mathbf{e}_{n2} = \mathbf{e}_{12} \times \mathbf{e}_{n1}$$

- 5 Fix the first point in all three directions, so that there are no rigid body translations.

$$\mathbf{u}_{p1} = 0$$

- 6 Constrain the second point  $p_2$  in two directions, so that all possible rotations except around the line  $\mathbf{e}_{12}$  are suppressed:

$$\mathbf{u}_{p2} \cdot \mathbf{e}_{n1} = 0$$

$$\mathbf{u}_{p2} \cdot \mathbf{e}_{n2} = 0$$

- 7 Constrain the third point  $p_3$  so that the remaining rotation is suppressed:

$$\mathbf{u}_{p3} \cdot \mathbf{e}_{n1} = 0$$

The body is now constrained against rigid body rotations, while still free to stretch in any direction.

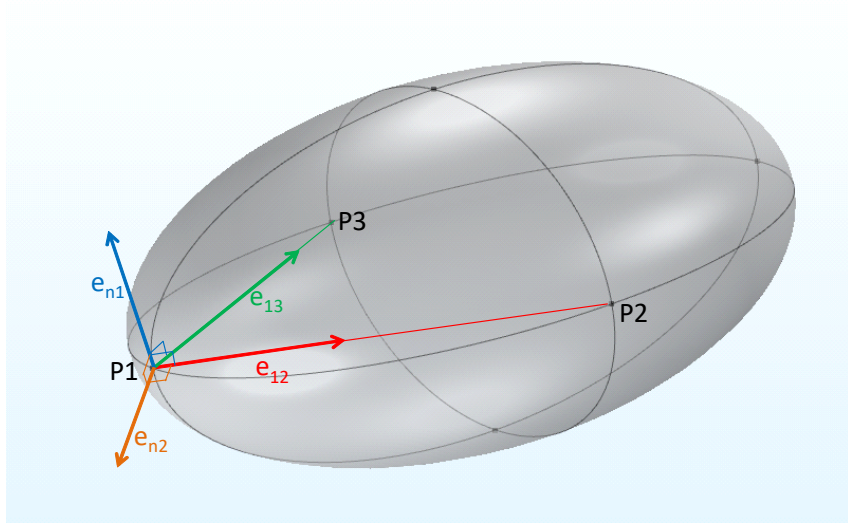


Figure 3-33: Selection of constraint orientations: P1 is fixed, P2 is constrained so that it can only move in the  $\mathbf{e}_{12}$  direction, and P3 is constrained in the  $\mathbf{e}_{n1}$  direction.

### 3D WITH ROTATIONAL DEGREES OF FREEDOM

In the Shell interface, the rigid body suppression is implemented using the translational degrees of freedom in three points, using the same algorithm as above.

In the Beam interface, a single point is constrained in all six degrees of freedom.

### 2D WITH TRANSLATIONAL DEGREES OF FREEDOM

In this case, which includes the Solid Mechanics, Multibody Dynamics, and Truss interfaces, three degrees of freedom must be constrained. As it is not possible to directly constrain the rotations, this must be done by a proper selection of locations and orientations for the constraints.

Two points, p1 and p2 are selected. The first point p1 is constrained in both directions in order to suppress translational motion. The second point is constrained in the direction orthogonal to the line joining the two points, so that rotation around the out-of-plane direction is suppressed:

$$\mathbf{u}_{p1} = 0$$

$$\mathbf{e}_{12} = \frac{\mathbf{X}_2 - \mathbf{X}_1}{|\mathbf{X}_2 - \mathbf{X}_1|}$$

$$\mathbf{u}_{p2} \cdot (\mathbf{e}_{12} \times \mathbf{e}_Z) = 0$$

## 2D WITH ROTATIONAL DEGREES OF FREEDOM

In the Plate interface, you can choose to use either three or six degrees of freedom. When six degrees of freedom are used, the same approach as for 3D solids and shells is used. In the case of three degrees of freedom, three points which are not located on a straight line are constrained against out-of-plane translation.

In the Beam interface, a single point is constrained in all three degrees of freedom.

## 2D AXIALLY SYMMETRIC

For both the Solid Mechanics and Membrane interfaces, a single point is constrained in the axial ( $Z$ ) direction.

### *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

$$\boldsymbol{\sigma} \cdot \mathbf{n} = -\rho c_p \left( \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n} \right) \mathbf{n} - \rho c_s \left( \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{t} \right) \mathbf{t}$$

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 is close to the normal at the wall.

In the general case, you can use

$$\boldsymbol{\sigma} \cdot \mathbf{n} = -\mathbf{d}_i(\rho, c_p, c_s) \frac{\partial \mathbf{u}}{\partial t}$$

where the mechanical impedance  $\mathbf{d}_i$  is a diagonal matrix available as the user input, and by default it is set to

$$\mathbf{d}_i = \rho \frac{c_p + c_s}{2} \mathbf{I}$$



More information about modeling using low-reflecting boundary conditions can be found in [Ref. 1](#).

### *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.

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

$$\mathbf{u}_{\text{destination}} = e^{-i\varphi} R_{\theta} \mathbf{u}_{\text{source}}^T$$

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  $\varphi$  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

$$\mathbf{F} = -\mathbf{F}_0 \exp[-im \operatorname{atan}(Y/X)]$$

when modeled using the Cartesian coordinates. The parameter  $m$  is often referred to as the *azimuthal wave number*.



Ref. 2 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 mesh nodes 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 mesh node. 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 located at an original coordinate  $\mathbf{X}$ , the normal orientation is computed as

$$\begin{aligned}\mathbf{X}' &= \mathbf{X} - \mathbf{X}_c \\ \mathbf{n}_r &= \mathbf{X}' - (\mathbf{X}' \cdot \mathbf{e}_s)\mathbf{e}_s\end{aligned}$$

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

$$\begin{aligned}\mathbf{X}' &= \mathbf{X} - \mathbf{X}_c \\ \mathbf{x}' &= \mathbf{x} - \mathbf{X}_c\end{aligned}$$

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 mesh node. The only input is a **Center of sphere**,  $\mathbf{X}_c$ . For a node 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 Domain**, **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

$$\varepsilon(\mathbf{X}) = \alpha(\mathbf{X}, T)(T - T_{ref})$$

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. 3](#), giving

$$u_i(\mathbf{X}) = u_i(\mathbf{X}_0) + \int_{\mathbf{X}_0}^{\mathbf{X}} \left( \varepsilon_{il} + (X_k - X_{k0}) \left( \frac{\partial \varepsilon_{il}}{\partial X_k} - \frac{\partial \varepsilon_{kl}}{\partial X_i} \right) \right) dX_l$$

Summation over double indices is implied. The rigid body rotation term given in [Ref. 3](#) 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 1 as

$$\mathbf{X}' = \mathbf{X}_0 + s\mathbf{p}$$

giving

$$u_i(\mathbf{X}) = \int_0^1 \left( \varepsilon_{il} + (1-s)p_k \left( \frac{\partial \varepsilon_{il}}{\partial X_k} - \frac{\partial \varepsilon_{kl}}{\partial X_i} \right) \right) p_l ds \quad (3-134)$$

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  $\Theta$  is given by

$$\Theta = \frac{1}{2} \nabla \times \mathbf{u}$$

Applying the rotation operator to [Equation 3-134](#) 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  $\varepsilon_{mni}$  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 Theory*

**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,

$$\mathbf{f} = \mathbf{n} \cdot \left\{ -p\mathbf{I} + \left( \mu(\nabla\mathbf{u}_{\text{fluid}} + (\nabla\mathbf{u}_{\text{fluid}})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u}_{\text{fluid}})\mathbf{I} \right) \right\} \quad (3-135)$$

where  $p$  denotes pressure,  $\mu$  the dynamic viscosity for the fluid,  $\mathbf{n}$  the outward normal to the boundary, 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 3-135](#), 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.



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. There one needs to use a homogeneous material model, but 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.

An RVE is typically identified as the smallest possible unit cell of a material. If the material has a random distribution of for example porosity, it 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 of them having the same geometry, but offset by a given distance.

To model an RVE, 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.

In principle, there are no limitations on the physics features you can use for modeling the RVE, as long as the basic assumptions about periodicity are not violated. 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 macro scale and a micro scale. The macro scale usually refers to the homogenized continuous media, and the micro scale to the heterogeneous unit cell, that is, the RVE. The macro stress tensor  $\bar{\sigma}$  and the macro strain tensor  $\bar{\varepsilon}$  are derived by averaging the stresses and strains in the periodic cell

$$\bar{\sigma} = \frac{1}{V} \int_V \sigma dV \quad \text{and} \quad \bar{\varepsilon} = \frac{1}{V} \int_V \varepsilon dV \quad (3-136)$$

where  $V$  is the volume of the cell. The macroscopic elasticity tensor of the homogenized continuum is then defined by

$$\bar{\sigma} = \bar{C} : \bar{\varepsilon} \quad (3-137)$$

### DISPLACEMENT CONTINUITY

For a periodic structure that consists of an array of repeated unit cells, the displacement field is written as

$$u = \bar{\epsilon}X + u^*$$

where  $\bar{\epsilon}$  is global average strain tensor, and  $u^*$  is a function that is periodic from one unit cell to another. As the array of cells is a continuous structure, displacement continuity must be satisfied across the boundaries between the cells. The boundaries in a **Boundary Pair** subnode must 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 boundary surfaces can be written as

$$u_{\text{dst}} = \bar{\epsilon}X_{\text{dst}} + u^* \quad \text{and} \quad u_{\text{src}} = \bar{\epsilon}X_{\text{src}} + u^*$$

Hence, the displacement continuity is established by

$$u_{\text{dst}} - u_{\text{src}} = \bar{\epsilon}(X_{\text{dst}} - X_{\text{src}}) \quad (3-138)$$

Together with displacement continuity, the traction continuity should be satisfied for the RVE, which is done implicitly in the displacement based finite element method. There are different options available to enforce the displacement continuity constraint:

- Free Expansion
- Prescribed Average Strain
- Prescribed Average Stress
- Mixed

### 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  $\bar{\epsilon}$  in Equation 3-138 is considered as a global degree of freedom that varies freely. The averaged coefficient of thermal expansion  $\alpha$  is computed as

$$\alpha = \frac{1}{V} \int_V \frac{\epsilon}{\Delta T} dV$$

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  $\beta$  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 3-138 as a constraint.

#### **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  $\bar{\varepsilon}$  in Equation 3-138 is a user input. The average stress is computed from Equation 3-136, and the elasticity tensor is obtained from Equation 3-137.

#### **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  $\bar{\varepsilon}$  in Equation 3-138 is a global degree of freedom, which is defined by a global weak equation

$$\bar{\sigma} : \delta \bar{\varepsilon} V \quad (3-139)$$

The compliance tensor is then obtained from the following equation

$$\bar{\varepsilon} = \bar{C}^{-1} : \bar{\sigma}$$

#### **MIXED**

This option studies the response of the unit cell when subjected to a combination of prescribed stress and strain. You can chose which components of the average stress and strain tensors to enter as user inputs; the remaining components are computed from Equation 3-136 and Equation 3-139 by enforcing Equation 3-138 as a constraint.



## Rigid Connector Theory

---

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

$$\mathbf{u}_r = (R - I) \cdot (\mathbf{X} - \mathbf{X}_c) + \mathbf{u}_c \quad (3-140)$$

The total rotation of the rigid connector can be also presented as a rotation vector. Its definition is

$$\Theta = \frac{2a \cos(a)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

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

$$(\mathbf{u} - \mathbf{u}_r) \cdot \text{test}(\mathbf{F}_{rs}) + \mathbf{F}_{rs} \cdot \text{test}(\mathbf{u} - \mathbf{u}_r) = 0 \quad (3-141)$$

Here  $\mathbf{u}$  is the displacement field on the boundaries,  $\mathbf{u}_r$  is the rigid body displacement as given by [Equation 3-140](#), 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 3-141. 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

$$\mathbf{F}_{rs} = \mathbf{F}_c + \mathbf{F}_d \times (\mathbf{X} - \mathbf{X}_c) \quad (3-142)$$

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 3-142 is replaced by

$$\mathbf{F}_{rs} = \mathbf{F}_c + \mathbf{F}_d \times (R \cdot (\mathbf{X} - \mathbf{X}_c)) \quad (3-143)$$

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> etc.  
Examples are `solid.rig1.Fc1x` or `solid.rig2.Fd5z`.

## Contact Analysis Theory

---

In COMSOL Multiphysics you can model contact between a group of boundaries in 2D or 3D. There are two classes of algorithms available: the *penalty* method and the *augmented Lagrangian* method.



- [Contact Modeling](#)
- Documentation of the [Contact](#), [Friction](#), [Slip Velocity](#), [Adhesion](#), [Decohesion](#), and [Wear](#) features
- [Identity and Contact 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 the 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 **Contact Pair** 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 3-34](#). For mechanical contact, most variables and equations are formulated on the destination boundary. Hence, the typical 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.

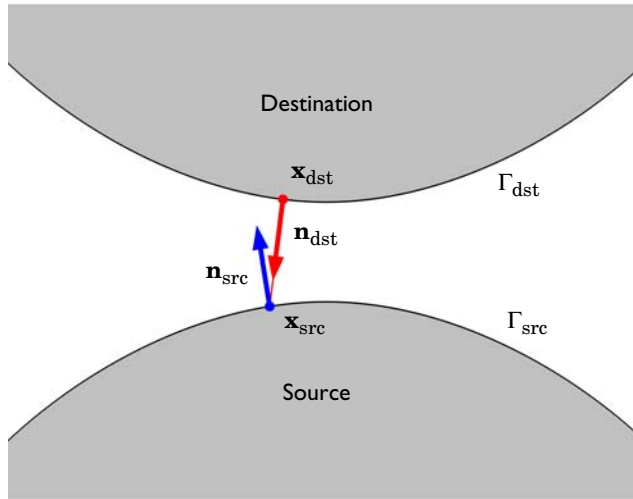


Figure 3-34: Schematic illustration of the contact search and mapping between a source and destination boundary.

For each point  $\mathbf{x}_{\text{dst}}$  in the spatial frame on the destination boundary  $\Gamma_{\text{dst}}$ , the closet point  $\mathbf{x}_{\text{src}}$  in the spatial frame on the source boundary  $\Gamma_{\text{src}}$  in the direction of the destination spatial normal  $\mathbf{n}_{\text{dst}}$  is sought. Such a mapping can be described through and operator  $\text{map}(\mathbf{E}, \mathbf{x})$ , where  $\mathbf{E}$  is some generic quantity or expression, and  $\mathbf{x}$  is the point to which  $\mathbf{E}$  is mapped. Using this mapping operator, the coordinates of the mapped point on  $\Gamma_{\text{src}}$  is

$$\mathbf{x}_{\text{src}} = \text{map}(\mathbf{x}, \mathbf{x}_{\text{dst}})$$

where  $\mathbf{x}$  is the spatial coordinates that parametrize  $\Gamma_{\text{src}}$ . 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  $\mathbf{x}_{\text{dst}}$  on  $\Gamma_{\text{dst}}$  if nothing else is specified.

An important quantity for contact mechanics is the gap function  $g_{\text{geom}}(\mathbf{x})$  defined as the geometric distance between the source and the destination boundary. Note that  $g_{\text{geom}}(\mathbf{x})$  is not necessarily the closest distance between the two boundaries. Using the mapping operator, the gap distance at  $\mathbf{x}_{\text{dst}}$  is defined as

$$g_{\text{geom}}(\mathbf{x}_{\text{dst}}) = -\mathbf{n}_{\text{dst}} \cdot [\mathbf{x}_{\text{dst}} - \text{map}(\mathbf{x})]$$

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 **Contact Pair** also defines a dedicated variable for the gap function with the name `geomgap_dst_<pair_tag>`.



- A **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>`.
- The mapping by default uses the spatial frame normal. If the **Mapping method** in the **Contact 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

$$g_n(\mathbf{x}_{dst}) = g_{geom} - d_{offset,d} - \frac{\text{map}(d_{offset,s})}{(\mathbf{n} \cdot \text{map}(\mathbf{n}))} \quad (3-144)$$

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 towards  $\mathbf{x}_{dst}$  as illustrated in [Figure 3-34](#). When the geometric surfaces are in contact, the normals will be exactly opposite.

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  $\Delta_{gap}$ . In this case, the physical gap function at  $\mathbf{x}_{dst}$  is

$$\mathbf{g}_n(\mathbf{x}_{dst}) = \begin{cases} g_{geom} - d_{offset,d} - \frac{\text{map}(d_{offset,s})}{-\mathbf{n} \cdot \text{map}(\mathbf{n})} & \text{if } g_{init} \geq \Delta_{gap} \\ g_{geom} - g_{init} - d_{offset,d} - \frac{\text{map}(d_{offset,s})}{-\mathbf{n} \cdot \text{map}(\mathbf{n})} & \text{else} \end{cases}$$

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  $\mathbf{g}_t$  between the two boundaries at  $\mathbf{x}_{dst}$  is defined as

$$\frac{\partial}{\partial t} \mathbf{g}_t(\mathbf{x}_{dst}) = \text{map}(\mathbf{F}) \text{map}\left(\frac{\partial \mathbf{X}}{\partial t}\right) \quad (3-145)$$

where  $\mathbf{X}$  is the material coordinates, and  $\mathbf{F}$  is the deformation gradient that contains information about the local rotation and stretch. 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 3-145 so that the incremental slip at  $\mathbf{x}_{dst}$  is

$$\Delta \mathbf{g}_t(\mathbf{x}_{dst}) = \text{map}(\mathbf{F})[\text{map}(\mathbf{X}) - \mathbf{X}_{src,old}]$$

where  $\mathbf{X}_{src,old}$  is the mapped material coordinates at the previous increment.



When a **Contact** node is added to the Shell, Layered Shell, or Membrane interfaces, the thickness of the structural element is included in the gap computation by considering the physical position of its top and bottom surface relative to the geometric boundary. The appropriate offset is then added to  $d_{offset,s}$  and  $d_{offset,d}$ . Similar corrections are also made when computing the incremental slip  $\Delta \mathbf{g}_t$ .

### 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 non-penetration condition. On the destination boundary  $\Gamma_{\text{dst}}$ , the normal contact condition can be written as a weak equation (the contribution of normal contact to the virtual work)

$$0 = \int_{\Gamma_{\text{dst}}} T_n \delta g_n dA \quad (3-146)$$

where  $T_n$  is the contact pressure, assumed positive in compression. To complete the contact condition, Equation 3-146 is subjected to the Kuhn–Tucker conditions

$$\begin{aligned} g_n &\geq 0 \\ T_n &\geq 0 \\ T_n g_n &= 0 \end{aligned} \quad (3-147)$$

where the first inequality represents the nonpenetration condition. Equation 3-146 and Equation 3-147 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 two main classes: the *penalty method* and the *augmented Lagrangian method*.

### The Penalty Method

For the penalty method, the contact pressure in Equation 3-146 is represented using a penalization of the physical gap so that

$$T_n = \begin{cases} -p_n g_n + p_0 & \text{if } g_n < p_0/p_n \\ 0 & \text{else} \end{cases} \quad (3-148)$$

where  $p_n$  is the penalty factor, and  $p_0$  is the pressure at zero gap. This leads to an approximative solution of the non-penetration condition. The penalty method as defined by Equation 3-148 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, it makes the problem ill-conditioned and unstable.

The pressure at zero gap,  $p_0$ , 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  $p_0 = 0$ , there must always be some overclosure in order for a contact pressure to develop.



In addition to [Equation 3-148](#), COMSOL Multiphysics also 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

$$T_n \frac{\partial g_n}{\partial t} = 0$$

In this case, the non-penetration and persistency conditions are enforced by penalizing the rate of the physical gap distance such that the contact pressure is defined by

$$T_n = \begin{cases} p_n \langle \frac{\partial g_n}{\partial t} \rangle & \text{if } g_{n, \text{old}} \leq 0 \\ 0 & \text{else} \end{cases} \quad (3-149)$$

where  $g_{n, \text{old}}$  is the physical gap distance at the previous increment, and  $\langle \rangle$  is the positive parts operator. With the definition in [Equation 3-149](#),  $T_n$  is only non-zero 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 3-148](#) and [Equation 3-149](#) simultaneously. In such a case, they are additive and conceptually represent a parallel coupling of a spring and a dashpot.

The weak contribution added for normal contact when using the penalty method is

$$0 = \dots + \int_{\Gamma_{\text{dst}}} T_n \delta g_n dA$$

#### *The Augmented Lagrangian Method*

The augmented Lagrangian method can be seen as a compromise between using the penalty method and enforcing the non-penetration 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 non-penetration condition exactly over the contacting boundaries. It is based on the following augmentation of [Equation 3-146](#)

$$0 = \dots + \int_{\Gamma_{\text{dst}}} T_{\text{np}} \delta g_n dA - \frac{1}{p_n} \int_{\Gamma_{\text{dst}}} (T_{\text{np}} - T_n) \delta T_n dA \quad (3-150)$$

where  $T_{\text{np}}$  is the penalized (or augmented) contact pressure, and  $T_n$  is a Lagrange multiplier that can be identified as a contact pressure. The Lagrange multiplier representation of the contact pressure 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, and this choice affects the definition of  $T_{\text{np}}$ .

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  $T_{\text{np}}$  is defined with a smooth transition to stabilize the problem when transitioning between contact states. The definition for  $T_{\text{np}}$  for outer iteration  $j+1$  is defined by

$$T_{\text{np},j+1} = \begin{cases} T_{n,j} - p_n g_n & \text{if } g_n \leq 0 \\ T_{n,j} e^{-\frac{p_n g_n}{T_n}} & \text{else} \end{cases}$$

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  $T_{n,j}^{n+1} = T_n^n$
- 2 Compute the displacement  $\mathbf{u}_{j+1}$  by solving

$$0 = \dots + \int_{\Gamma_{\text{dst}}} T_{\text{np},j+1} \delta g_{n,j+1} dA$$

- 3 Compute the Lagrange multiplier  $T_{n,j+1}$  by solving

$$0 = \int_{\Gamma_{\text{dst}}} (T_{\text{np},j+1} - T_{n,j+1}) \delta T_n dA$$

- 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 3-150 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

$$T_{np,j+1} = \begin{cases} T_{n,j} - p_n g_n & \text{if } g_n \leq 0 \\ 0 & \text{else} \end{cases}$$

The **Fully Coupled** method can be convenient as it enforces the non-penetration 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 non-penetration and persistency conditions are enforced by penalizing the rate of the physical gap distance so that

$$T_{np,j+1} = \begin{cases} \langle T_{n,j} - p_n \frac{\partial g_n}{\partial t} \rangle & \text{if } g_{n,old} \leq 0 \\ 0 & \text{else} \end{cases} \quad (3-151)$$

where  $g_{n,old}$  is the physical gap distance at the previous increment, and  $\langle \rangle$  is the positive parts operator. The same segregated solution strategy as described above is applied also when the dynamic method is used.

### *Dissipation*

The **Penalty, dynamic** and **Augmented Lagrange, dynamic** methods are based on a viscous type formulation, and dissipate energy when the contact is active. The accumulated dissipated energy  $W_{\text{cntv}}$  can be computed by solving a distributed ODE on the destination boundary:

$$\frac{\partial}{\partial t} W_{\text{cntv}} = T_n \frac{\partial g_n}{\partial t}$$

### **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  $\Gamma_{\text{dst}}$ , the tangential contact condition can be written as a weak equation (the contribution of tangential contact to the virtual work)

$$0 = \int_{\Gamma_{\text{dst}}} \mathbf{T}_t \cdot \delta \mathbf{g}_t dA \quad (3-152)$$

where  $\mathbf{T}_t$  is the friction force. The stick and slip conditions are then enforced by the constitutive model used to define  $\mathbf{T}_t$ .

### *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

$$\begin{aligned} F_{\text{slip}} &\equiv \|\mathbf{T}_t\| - T_{t, \text{crit}} \leq 0 \\ \frac{\partial \mathbf{g}_t}{\partial t} &= \lambda \frac{\mathbf{T}_t}{\|\mathbf{T}_t\|} \\ \lambda &\geq 0 \\ \lambda F_{\text{slip}} &= 0 \end{aligned} \quad (3-153)$$

where  $F_{\text{slip}}$  is the slip function (analogue to the yield function in plasticity theory), and  $\lambda$  is friction multiplier. The critical friction force  $T_{t, \text{crit}}$  determines when slip occurs, and is defined as

$$T_{t, \text{crit}} = \min(T_{\text{cohe}} + \mu T_n, T_{\text{cohe}})$$

where  $T_{\text{cohe}}$  is the cohesion, and  $T_{t, \text{max}}$  defines the maximum friction force that is admissible. The friction coefficient  $\mu$  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

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}}) \exp(-\alpha_{\text{def}} \|\mathbf{v}_s\|)$$

where  $\mu_{\text{stat}}$  is the static friction coefficient,  $\mu_{\text{dyn}}$  is the dynamic friction coefficient,  $\mathbf{v}_s$  is the slip velocity, and  $\alpha_{\text{def}}$  is a decay coefficient.

It is possible to also define the critical friction force through an arbitrary expression. In principle,  $T_{t, \text{crit}}$ , 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 depend on  $\mathbf{T}_t$ . 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 3-153 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  $n+1$ :

$$\begin{aligned} F_{\text{slip}, n+1} &\equiv \|\mathbf{T}_{t, n+1}\| - T_{t, \text{crit}, n+1} \leq 0 \\ \mathbf{T}_{t, n+1} &= \mathbf{T}_{t, n} + p_t \left( \Delta \mathbf{g}_t - \lambda \frac{\mathbf{T}_{t, n+1}}{\|\mathbf{T}_{t, n+1}\|} \right) \\ \lambda &\geq 0 \\ \lambda F_{\text{slip}, n+1} &= 0 \end{aligned}$$

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  $\lambda$  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

$$\mathbf{T}_{t, n+1} = \begin{cases} \mathbf{T}_{t, n+1}^{\text{trial}} & \text{if } F_{\text{slip}, n+1} \leq 0 \\ T_{t, \text{crit}, n+1} \frac{\mathbf{T}_{t, n+1}^{\text{trial}}}{\|\mathbf{T}_{t, n+1}^{\text{trial}}\|} & \text{if } F_{\text{slip}, n+1} > 0 \end{cases}$$

The weak contribution added for tangential contact when using the penalty method follows from Equation 3-152, and is given by

$$0 = \dots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{t, n+1} \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

#### *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 3-152

$$0 = \dots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{tp}} \delta \mathbf{g}_t dA + \frac{1}{p_t} \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{\text{tp}} - \mathbf{T}_t) \delta \mathbf{T}_t dA \quad (3-154)$$

where  $\mathbf{T}_{\text{tp}}$  is the penalized (or augmented) friction, and  $\mathbf{T}_t$  is a Lagrange multiplier that can be identified as a traction. The Lagrange multiplier representation of the tangential traction 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 3-153 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

$$\mathbf{T}_{t, j+1}^{\text{trial}} = \mathbf{T}_{t, j} - p_t \Delta \mathbf{g}_t$$

and the return mapping gives the current value of the penalized friction force

$$\mathbf{T}_{\text{tp},j+1} = \begin{cases} \mathbf{T}_{\text{t},j+1}^{\text{trial}} & \text{if } F_{\text{slip},j+1} \leq 0 \\ T_{\text{t,crit},j+1} \frac{\mathbf{T}_{\text{t},i+1}^{\text{trial}}}{\|\mathbf{T}_{\text{t},j+1}^{\text{trial}}\|} & \text{if } F_{\text{slip},j+1} > 0 \end{cases}$$

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  $\mathbf{T}_{\text{t},j}^{n+1} = \mathbf{T}_{\text{t}}^n$

2 Compute the displacement  $\mathbf{u}_{j+1}$  by solving

$$0 = \dots + \int_{\Gamma_{\text{dst}}} T_{\text{np},j+1} \delta \mathbf{g}_n dA + \int_{\Gamma_{\text{dst}}} \mathbf{T}_{\text{tp},j+1} \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

3 Compute the Lagrange multipliers  $T_{n,j+1}$  and  $\mathbf{T}_{\text{t},j+1}$  by solving

$$0 = \int_{\Gamma_{\text{dst}}} (T_{\text{np},j+1} - T_{n,j+1}) \delta T_n dA + \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{\text{tp},j+1} - \mathbf{T}_{\text{t},j+1}) \delta T_n dA$$

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

$$\mathbf{T}_{\text{t},i+1}^{\text{trial}} = \mathbf{T}_{\text{t},i} - p_t \Delta \mathbf{g}_t$$

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 3-154](#), and is

$$0 = \dots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_t \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA + \frac{1}{p_t} \int_{\Gamma_{\text{dst}}} (\mathbf{T}_{t_p} - \mathbf{T}_t) \delta \mathbf{T}_t dA$$

#### *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

$$\frac{\partial \mathbf{g}_t}{\partial t} = \mathbf{v}_{\text{slip}}$$

where  $\mathbf{v}_{\text{slip}}$  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 3-153](#) is simplified, and it follows that

$$\mathbf{T}_t = T_{t, \text{crit}} \frac{-\mathbf{v}_{\text{slip}}}{\|\mathbf{v}_{\text{slip}}\|}$$

since the slip velocity and friction force are co-linear. The weak contribution added for tangential contact by the slip velocity follows from [Equation 3-152](#) and is

$$0 = \dots + \int_{\Gamma_{\text{dst}}} \mathbf{T}_t \cdot \text{map}(\mathbf{F}) \delta[\text{map}(\mathbf{X})] dA$$

#### *Dissipation*

Friction is a dissipative process and the accumulated dissipated energy  $W_{\text{fric}}$  can be computed by solving a distributed ODE on the destination boundary:

$$\frac{\partial}{\partial t} W_{\text{fric}} = -\mathbf{T}_t \cdot \frac{\partial \mathbf{g}_t}{\partial t}$$

#### **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.

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  $\mathbf{T}_b^{-T}$  contains the transform 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  $n_\tau$  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

$$\int_{dst} (\mathbf{f} \cdot \delta \mathbf{u}) dA$$

### 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

$$\mathbf{f} = (1 - d)\mathbf{k}\mathbf{u} = \mathbf{k}\mathbf{u} - d\mathbf{k}\mathbf{u} \quad (3-155)$$

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 3-36](#). 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 3-155](#), 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.

$$u_m = \|\mathbf{u}\|$$

To keep track of the current state of damage, the maximum value of  $u_m$  over the loading history is defined as

$$u_{m, \max} = \max(u_m, u_{m, \max}^{\text{old}})$$

where  $u_{m, \max}^{\text{old}}$  is an internal degree of freedom that takes the value of  $u_{m, \max}$  at the previous converged solution. The damage variable is then defined as a function of  $u_{m, \max}$  of the form

$$d = \begin{cases} 0 & \text{if } u_{m, \max} < u_{0m} \\ \min[F^{-1}(u_{m, \max}), 1] & \text{else} \end{cases}$$

where  $F^{-1}$  is called the damage evolution function and  $u_{0m}$  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  $F^{-1}$  are available in the **Traction separation law** list. These damage evolution functions are summarized in [Figure 3-35](#), and the resulting traction separations laws in [Figure 3-36](#).

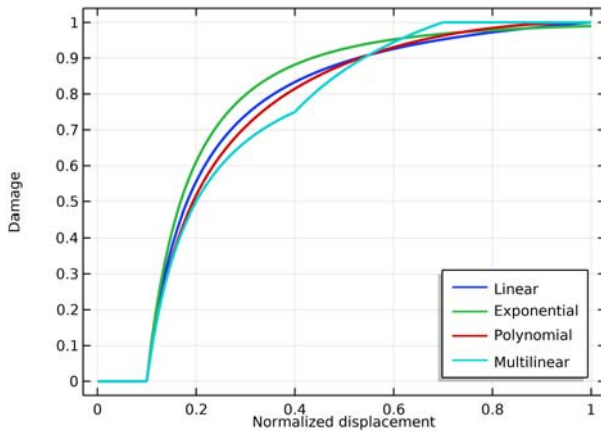


Figure 3-35: Damage evolution functions available with the displacement-based damage model.

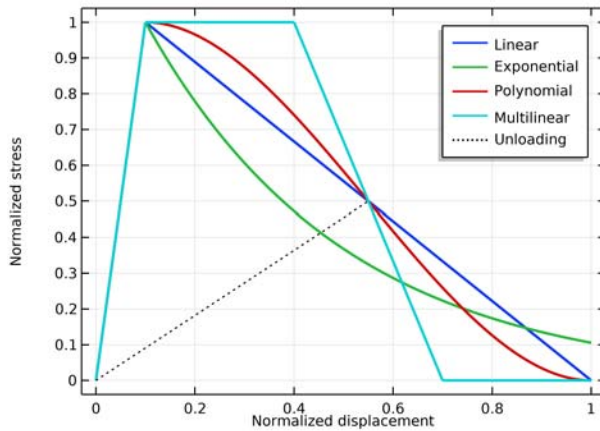


Figure 3-36: Traction separation laws available with the displacement-based damage model.

The **Linear** option specifies a damage evolution function that gives a bilinear traction separation law as seen in Figure 3-36. It is defined as

$$F^{-1}(u_{m, \max}) = \frac{u_{fm}}{u_{m, \max}} \left( \frac{u_{m, \max} - u_{0m}}{u_{fm} - u_{0m}} \right)$$

where  $u_{0m}$  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

$$u_{0m} = u_{0t} u_{0s} \sqrt{\frac{u_m^2}{\langle u_I \rangle^2 u_{0s}^2 + u_{II}^2 u_{0t}^2}} \quad (3-156)$$

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  $u_{0t}$  and  $u_{0s}$  are calculated as

$$u_{0t} = \frac{\sigma_t}{k_n} \quad \text{and} \quad u_{0s} = \frac{\sigma_s}{k_t} \quad (3-157)$$

where  $\sigma_t$  is the tensile strength and  $\sigma_s$  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

$$\left( \frac{G_I}{G_{ct}} \right)^\alpha + \left( \frac{G_{II}}{G_{cs}} \right)^\alpha = 1$$

and the **Benzeggagh-Kenane** criterion is defined as

$$G_{ct} + (G_{cs} - G_{cs}) \left( \frac{G_{II}}{G_I + G_{II}} \right)^\alpha = G_I + G_{II}$$

where  $G_{ct}$  and  $G_{cs}$  are the tensile and shear energy release rates, respectively. The exponent  $\alpha$  is called the mode mixity exponent. From these relations, the mixed mode failure displacement can be calculated. For the power law criterion, the expression is

$$u_{mf} = \begin{cases} \frac{2(1+\beta^2)}{u_{m0}} \left[ \left( \frac{k_n}{G_{ct}} \right)^\alpha + \left( \frac{\beta^2 k_t}{G_{cs}} \right)^\alpha \right]^{-\frac{1}{\alpha}} & \text{if } u_I > 0 \\ \frac{2G_{cs}}{\sigma_s} & \text{if } u_I \leq 0 \end{cases} \quad (3-158)$$

with  $\beta = u_{II}/u_I$ . For the Benzeggagh–Kenane criterion, the corresponding expression is

$$u_{mf} = \begin{cases} \frac{2u_m^2}{u_{m0}(k_n u_I^2 + k_t u_{II}^2)} \left[ G_{ct} + (G_{cs} - G_{ct}) \left( \frac{\frac{k_t}{k_n} u_{II}^2}{u_I^2 + \frac{k_t}{k_n} u_{II}^2} \right)^\alpha \right] & \text{if } u_I > 0 \\ \frac{2G_{cs}}{\sigma_s} & \text{if } u_I \leq 0 \end{cases} \quad (3-159)$$

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 reaches zero asymptotically as seen in [Figure 3-36](#).

$$F^{-1}(u_{m, \max}) = 1 - \frac{u_{0m}}{u_{m, \max}} \exp\left(-\frac{u_{m, \max} - u_{0m}}{u_{fm} - u_{0m}}\right)$$

The mixed mode damage initiation displacement  $u_{0m}$  is defined by [Equation 3-156](#) and [Equation 3-157](#). The mixed mode failure displacement  $u_{fm}$  is for the power law again given by

$$u_{mf} = \begin{cases} u_{m0}(1 + \beta^2) \left[ \left( k_n \frac{u_{m0}^2}{G_{ct}} \right)^\alpha + \left( k_t \frac{\beta^2 u_{m0}^2}{G_{cs}} \right)^\alpha \right]^{\frac{-1}{\alpha}} & \text{if } u_I > 0 \\ \frac{G_{cs}}{\sigma_s} & \text{if } u_I \leq 0 \end{cases}$$

and for the Benzeggagh–Kenane criterion by

$$u_{mf} = \begin{cases} \frac{u_m^2}{u_{m0}(k_n u_I^2 + k_t u_{II}^2)} \left[ G_{ct} + (G_{cs} - G_{ct}) \left( \frac{\frac{k_t}{k_n} u_{II}^2}{u_I^2 + \frac{k_t}{k_n} u_{II}^2} \right)^\alpha \right] & \text{if } u_I > 0 \\ \frac{G_{cs}}{\sigma_s} & \text{if } u_I \leq 0 \end{cases}$$

The **Polynomial** option specifies a damage evolution function that gives a traction separation law that is linear up to the interface strength, and thereafter softens with a cubic polynomial curve as seen in [Figure 3-36](#).

$$F^{-1}(u_{m, \max}) = 1 - \frac{u_{0m}}{u_{m, \max}} \left[ 1 + \left( \frac{u_{m, \max} - u_{0m}}{u_{fm} - u_{0m}} \right)^2 \left( 2 \frac{u_{m, \max} - u_{0m}}{u_{fm} - u_{0m}} - 3 \right) \right]$$

The mixed mode damage initiation displacement  $u_{0m}$  is defined by [Equation 3-156](#) and [Equation 3-157](#), and the mixed mode failure displacement  $u_{fm}$  by [Equation 3-158](#) or [Equation 3-159](#).

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 3-36](#).

$$F^{-1}(u_{m, \max}) = \begin{cases} 1 - \frac{u_{0m}}{u_{m, \max}} & \text{if } u_{m, \max} < u_{pm} \\ 1 - \frac{u_{0m}}{u_{m, \max}} \left( \frac{u_{fm} - u_{m, \max}}{u_{fm} - u_{pm}} \right) & \text{if } u_{m, \max} \geq u_{pm} \end{cases}$$

The mixed mode damage initiation displacement  $u_{0m}$  is defined by [Equation 3-156](#) and [Equation 3-157](#). The new constant  $u_{pm}$  defines the end of the region of constant stress and requires the introduction of the shape factor  $\lambda$ . The shape factor defines the ratio between the constant stress part of  $G_{ct}$  and the total “inelastic” part of  $G_{ct}$ :

$$\lambda = \frac{\sigma_t(u_{pt} - u_{0t})}{\sigma_t(u_{pt} - u_{0t}) + \frac{1}{2}\sigma_t(u_{ft} - u_{pt})} = \frac{\sigma_t(u_{pt} - u_{0t})}{G_{ct} - \frac{1}{2}\sigma_t u_{0t}}$$

Note that the shape factor is similarly defined for shear and 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  $u_{pi}$  for the respective component can be expressed as

$$u_{pi} = \lambda \frac{G_{ci}}{\sigma_i} + u_{0i} \left( 1 - \frac{\lambda}{2} \right)$$

and the failure displacement  $u_{fi}$  as

$$u_{fi} = (2 - \lambda) \frac{G_{ci}}{\sigma_i} + \frac{\lambda u_{0i}}{2}$$

where 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 and failure displacements are given as

$$u_{pm} = \sqrt{\frac{u_{m, \max}^2}{u_{I}^2 u_{ps}^2 + u_{II}^2 u_{pt}^2}}$$

$$u_{fm} = 2G_{ct}G_{cs} \frac{1 + \beta^2}{u_{0m}(k_n G_{cs} + \beta^2 k_t G_{cs})} - (u_{pm} - u_{0m})$$

#### *Energy-based damage*

The energy-based damage models provide an alternative formulation that relates the growth of damage to the dissipated mechanical energy of the interface. The formulation is based on the work presented in Ref. 4 and Ref. 5. The derivation starts from a stored energy function

$$\psi(\mathbf{u}, d) = \frac{1}{2}(1 - d)\mathbf{k}\mathbf{u} \cdot \mathbf{u}$$

From  $\psi(\mathbf{u}, d)$ , the stress vector  $\mathbf{f}$  and damage energy release rate  $Y_{dm}$  are obtained as

$$\mathbf{f} = \frac{\partial(\psi(\mathbf{u}, d))}{\partial \mathbf{u}} = (1 - d)\mathbf{k}\mathbf{u}$$

$$Y_{dm} = -\frac{\partial(\psi(\mathbf{u}, d))}{\partial d} = \frac{1}{2}\mathbf{k}\mathbf{u} \cdot \mathbf{u}$$

To keep track of the current state of damage, the maximum value of  $Y_d$  over the loading history is defined as

$$Y_{dm, \max} = \max(Y_{dm}, Y_{dm, \max}^{\text{old}})$$

where  $Y_{d, \max}^{\text{old}}$  is an internal degree of freedom that takes the value of  $Y_{dm, \max}$  at the previous converged solution. The energy dissipated during the decohesion process is

$$\int_0^{\infty} \dot{Y}_{dm, \max} dt = G_{cm}$$



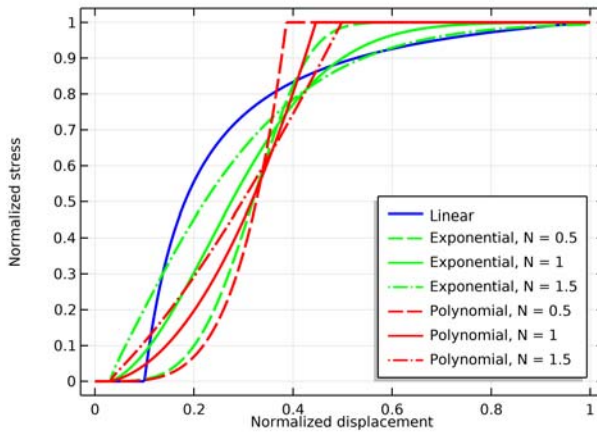
where  $G_{cm}$  is the critical energy release rate in the sense of fracture mechanics. The overall behavior of the cohesive zone model is then summarized by

$$Y_{dm, \max} = \begin{cases} Y_{d0m} & \text{if } d = 0 \\ t \\ \int \dot{Y}_{dm, \max} dt = F(d) & \text{if } 0 < d < 1 \\ 0 & \\ Y_{dm} & \text{if } d = 1 \end{cases}$$

where  $Y_{d0m}$  defines the damage threshold and  $F$  is a monotonically increasing function of the damage variable. From the above, an expression for the damage variable is obtained as

$$d = \begin{cases} 0 & Y_{dm, \max} < Y_{d0m} \\ \min[F^{-1}(Y_{dm, \max}), 1] & \text{else} \end{cases}$$

Three different definitions of  $F^{-1}$  are available under the **Traction separation law** list for the energy-based damage model. These damage evolution functions are summarized in [Figure 3-37](#), and the resulting traction separations laws in [Figure 3-38](#).



*Figure 3-37: 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.*

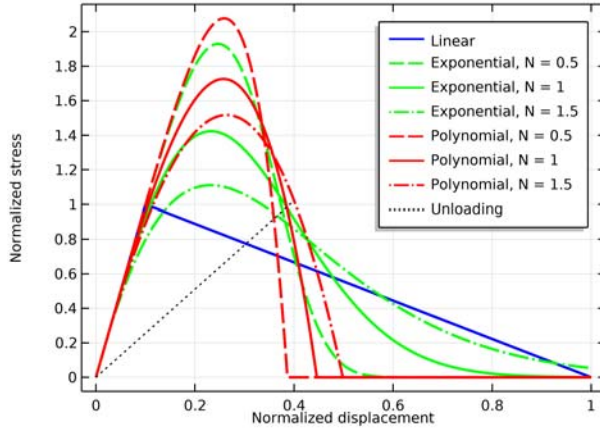


Figure 3-38: Traction separation laws available with the energy-based damage model. Note that the initial damage threshold is 10 times lower for the Exponential and Polynomial laws.

The **Linear** option specifies a damage evolution function that gives a bilinear traction separation law as seen in Figure 3-38.

$$F^{-1}(Y_{d, \max}) = \frac{Y_{dfm}}{Y_{d0m} - Y_{dfm}} \left( \sqrt{\frac{Y_{d0m}}{Y_{d, \max}}} - 1 \right)$$

For the linear law, the equation constant  $Y_{dfm} = G_{cm}$ . The **Exponential** option specifies a more general damage evolution function of the following form

$$F^{-1}(Y_{d, \max}) = 1 - \exp\left(\frac{Y_{d, \max} - Y_{d0m}}{Y_{dfm} - Y_{d0m}}\right)^{\frac{1}{N}}$$

where  $N$  is a smoothening parameter with a default value equal to 1. The effect of  $N$  on the traction separation curve can be seen in Figure 3-38. For the exponential law,  $Y_{dfm}$  is defined as

$$Y_{dfm} = Y_{d0m} + \frac{G_{cm} - Y_{d0m}}{\Gamma(1 + N)}$$

where  $\Gamma()$  is the gamma function. A similar damage evolution function is obtained by the **Polynomial** option. It gives a generalized version of the traction separation law proposed in Ref. 5 of the form

$$F^{-1}(Y_{d, \max}) = \left( \frac{Y_{d, \max} - Y_{d0m}}{Y_{dfm} - Y_{d0m}} \right)^{\frac{1}{N}}$$

where

$$Y_{dfm} = N \left[ \left( \frac{1}{N} + 1 \right) G_{cm} - Y_{d0m} \right]$$

Left to determine are the two variables  $Y_{d0m}$  and  $G_{cm}$ . Their definitions require the introduction of a few new concepts related to the mixed mode loading. First, a mixed mode ratio is introduced

$$\beta = \sqrt{\frac{k_t}{k_n}} \left( \frac{u_{II}}{u_I} \right)$$

where  $u_I$  and  $u_{II}$  are the mode I and mode II displacements, respectively. These are obtained from the displacement jump vector  $\mathbf{u}$ . The energy release rates of the respective modes are then given by

$$Y_{dI} = \frac{1}{1 + \beta^2} Y_{dm} \quad \text{and} \quad Y_{dII} = \frac{\beta^2}{1 + \beta^2} Y_{dm}$$

Next, 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

$$\left( \frac{Y_{dI}}{G_{0t}} \right)^{\alpha_0} + \left( \frac{Y_{dII}}{G_{0s}} \right)^{\alpha_0} = 1$$

where  $G_{0t}$  and  $G_{0s}$  define the damage threshold in tension and shear, respectively. The corresponding criterion for the fracture toughness is

$$\left( \frac{G_I}{G_{ct}} \right)^{\alpha_c} + \left( \frac{G_{II}}{G_{cs}} \right)^{\alpha_c} = 1$$

where  $G_{ct}$  and  $G_{cs}$  are the critical energy release rates for tension and shear, respectively. The variables  $\alpha_0$  and  $\alpha_c$  are called mode mixity exponents. Using these mode mixture rules results in

$$Y_{d0m} = \frac{(1 + \beta^2)G_{0t}G_{0s}}{\left\{ G_{0s}^{\alpha_0} + (\beta^2 G_{0t})^{\alpha_0} \right\}^{\frac{1}{\alpha_0}}}$$

$$G_{cm} = \frac{(1 + \beta^2)G_{ct}G_{cs}}{\left\{ G_{cs}^{\alpha_c} + (\beta^2 G_{ct})^{\alpha_c} \right\}^{\frac{1}{\alpha_c}}}$$

### 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 instability leading to nonconverging solutions, or bifurcation causing 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 adds a viscous delay to the damage growth, and is available with time-dependent studies. The formulation introduces a viscous damage variable  $d_v$ , and redefines Equation 3-155 so that

$$\mathbf{f} = (1 - d_v)\mathbf{k}^T \mathbf{u} = \mathbf{k}^T \mathbf{u} - d_v \mathbf{k}^T \mathbf{u}$$

The variation of the viscous damage variable is defined as

$$\dot{d}_v = -\frac{1}{\tau}(d_v - d) \quad (3-160)$$

where  $d$  is the damage variable obtained from any of the available CZM, and  $\tau$  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  $\tau$  must be chosen with care. As a rule of thumb,  $\tau$  should at least be one or two orders of magnitude smaller than the expected time step. Too large values of  $\tau$  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 from numerical integration of Equation 3-160, 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  $\Delta t$  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 degree of freedom.

#### Damage Dissipation

The dissipation due to damage is given as

$$\dot{W}_{\text{dmg}} = \dot{d}W_{\text{sd}} = \dot{d}\left(\frac{1}{2}\mathbf{k}^T \mathbf{u} \cdot \mathbf{u}\right)$$

The energy dissipated due to damage at the current step is approximated by

$$W_{\text{dmg}}^{n+1} = (d^{n+1} - d^n) \left( \frac{W_{\text{sd}}^{n+1} - W_{\text{sd}}^n}{2} + W_{\text{sd}}^{n+1} \right) + W_{\text{dmg}}^n$$

where the required values at the previously converged step  $n$  are stored as internal degrees of freedom.

#### 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.

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  $h_{\text{wear}}$ . 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  $h_{\text{wear}}$  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,  $h_{\text{wear}}$  is subtracted from the offset variables  $d_{\text{offset},s}$  and  $d_{\text{offset},d}$  in Equation 3-144. 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.

The accumulated wear is computed from an evolution equation of the following general format

$$\frac{\partial h_{\text{wear}}}{\partial t} = f(\mathbf{v}_{\text{slip}}, T_{\text{N}}, T, \theta)$$

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  $T_{\text{n}}$ , and the temperature  $T$ . The surface and material properties also play an important role, and are represented by the generic quantity  $\theta$  in the above equation.

In COMSOL Multiphysics, the wear depth can be computed using a **Generalized Archard** law for which the wear rate is

$$\frac{\partial h_{\text{wear}}}{\partial t} = k_{\text{wear}} \left( \frac{T_{\text{n}}}{T_{\text{n,ref}}} \right)^n \|\mathbf{v}_{\text{slip}}\| \quad (3-161)$$

Here  $k_{\text{wear}}$  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 3-161 by setting  $n = 1$  and  $T_{\text{n,ref}} = 1$  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  $T_{\text{n}}$  and  $\mathbf{v}_{\text{slip}}$ , 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 3-161 actually reads

$$\frac{\partial h_{\text{wear}}}{\partial t} = k_{\text{wear}} \left[ \frac{\text{map}(T_n, \mathbf{x}_{\text{src}})}{T_{n,\text{ref}}} \right]^n \text{map}(\|\mathbf{v}_{\text{slip}}\|, \mathbf{x}_{\text{src}})$$

The definition of the wear rate thus includes multiple mappings from source to destination, and from 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.

# 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:

$$\sigma_{m,ij} = \frac{1}{L} \int_0^L \sigma_{ij} dx_1$$

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.

$$\sigma_{b,ij} = \left(1 - \frac{2x_1}{L}\right) \sigma_{b(\max),ij}$$

The maximum bending stress is defined so that the linear stress distribution has the same moment as the true stress distribution.

$$\sigma_{b(\max),ij} = \frac{6}{L^2} \int_0^L \sigma_{ij} \left(\frac{L}{2} - x_1\right) dx_1$$

The linearized stress distribution is the sum of the membrane and bending stresses,

$$\sigma_{mb,ij}(x_1) = \sigma_{m,ij} + \sigma_{b,ij}(x_1)$$



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.

$$\begin{aligned}\sigma_{p(\text{start}),ij} &= [\sigma_{ij} - \sigma_{mb,ij}]_{x_1=0} = [\sigma_{ij}]_{x_1=0} - \sigma_{m,ij} + \sigma_{b(\text{max}),ij} \\ \sigma_{p(\text{end}),ij} &= [\sigma_{ij} - \sigma_{mb,ij}]_{x_1=L} = [\sigma_{ij}]_{x_1=L} - \sigma_{m,ij} - \sigma_{b(\text{max}),ij}\end{aligned}$$

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

$$\sigma_{\text{int}} = \sigma_{p1} - \sigma_{p3}$$

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 at the endpoints are thus computed using the following stress tensor:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} \sigma_{m,11} \\ \sigma_{mb,22} \\ \sigma_{mb,33} \\ \sigma_{m,12} \\ \sigma_{m,13} \\ \sigma_{mb,23} \end{bmatrix}$$

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

$$\begin{bmatrix} N_{22} \\ N_{33} \\ N_{23} \end{bmatrix} = L \begin{bmatrix} \sigma_{m,22} \\ \sigma_{m,33} \\ \sigma_{m,23} \end{bmatrix}$$

The bending moments are computed from the bending stresses as

$$\begin{bmatrix} M_{22} \\ M_{33} \\ M_{23} \end{bmatrix} = \frac{L^2}{6} \begin{bmatrix} \sigma_{b(\text{max}),22} \\ \sigma_{b(\text{max}),33} \\ \sigma_{b(\text{max}),23} \end{bmatrix}$$

The out-of-plane shear forces are computed from the membrane stresses as

$$\begin{bmatrix} Q_2 \\ Q_3 \end{bmatrix} = L \begin{bmatrix} \sigma_{m,12} \\ \sigma_{m,13} \end{bmatrix}$$

# 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

$$J = \int_{\Gamma_{\text{tot}}} \left[ W_s dy - (\boldsymbol{\sigma} \cdot \mathbf{m}) \cdot \frac{\partial \mathbf{u}}{\partial x} ds \right] \quad (3-162)$$

where  $W_s$  is the strain energy density,  $\boldsymbol{\sigma}$  is a stress tensor, and  $\mathbf{m}$  is the outward normal of the integration contour  $\Gamma$ . In Equation 3-162, the crack is assumed to extend in the positive  $x$ -direction, and integration is made over  $\Gamma_{\text{tot}} = \Gamma \cup \Gamma_{\text{face}}$ , which can be any closed path around crack tip, see Figure 3-39. However, if there is no loading on the crack face, the contribution to  $J$  on  $\Gamma_{\text{face}}$  is zero, and it is sufficient to only perform the integration over  $\Gamma$ .

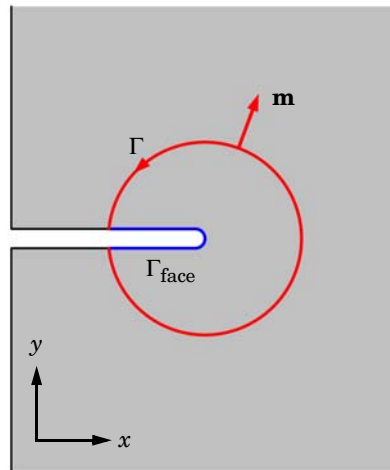


Figure 3-39: Circular J-integral around a notch in two dimensions.

For a general crack extension direction  $\mathbf{p}$ , Equation 3-162 can be generalized to

$$J = \int_{\Gamma} [W_s(\mathbf{m} \cdot \mathbf{p}) - (\boldsymbol{\sigma} \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] ds$$

In 3D, the path integral can be taken around an arbitrary point of the crack front, and  $\Gamma_{\text{tot}}$  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  $\Gamma_{\text{tot}}$ . The expression is then further extended to

$$J = \int_{\Gamma} [W_s(\mathbf{m} \cdot \mathbf{p}) - (\boldsymbol{\sigma} \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] ds + \int_A \nabla [(\boldsymbol{\sigma} \cdot \mathbf{t}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] \cdot \mathbf{t} dA$$

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  $r$ - $z$  plane, and  $\mathbf{t}$  is oriented in the azimuthal direction. The surface integral can then be simplified, so that

$$J = \int_{\Gamma} [W_s(\mathbf{m} \cdot \mathbf{p}) - (\boldsymbol{\sigma} \cdot \mathbf{m}) \cdot (\nabla \mathbf{u} \cdot \mathbf{p})] ds + (\mathbf{e}_r \cdot \mathbf{p}) \int_A [W_s - \sigma_{\phi\phi} (\nabla \mathbf{u})_{\phi\phi}] dA$$

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 non-zero. This additional contribution,  $J_{\text{face}}$ , is defined by

$$J_{\text{face}} = - \int_{\Gamma_{\text{face}}} \mathbf{F}_A \cdot (\nabla \mathbf{u} \cdot \mathbf{p}) ds$$

where  $\Gamma_{\text{face}}$  is schematically defined in [Figure 3-39](#).

### STRESS INTENSITY FACTORS

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

$$J = J_I + J_{II} + J_{III} = \frac{1}{E_{\text{eff}}} \left( K_I^2 + K_{II}^2 + \frac{1}{1-\nu} K_{III}^2 \right) \quad (3-163)$$

where  $E_{\text{eff}}$  is an effective Young's modulus that accounts for the stress state at the crack front. Assuming that  $K_{II} = \beta K_I$  and  $K_{III} = \gamma K_I / (1 + \nu)$ , where  $\beta$  and  $\gamma$  are coefficients that account for the mode mixture, it follows from Equation 3-163 that the stress intensity factors are

$$\begin{aligned} K_I &= \sqrt{\frac{E_{\text{eff}}}{1 + \beta^2 + \frac{1}{1+\nu}\gamma^2} J} \\ K_{II} &= \sqrt{\frac{E_{\text{eff}}}{1 + \frac{1}{\beta^2} + \frac{1}{1+\nu}\frac{\gamma^2}{\beta^2}} J} \\ K_{III} &= \sqrt{\frac{E_{\text{eff}}}{(1+\nu) \left[ 1 + \frac{1+\nu}{\gamma^2} + \frac{(1+\nu)\beta^2}{\gamma^2} \right]} J} \end{aligned} \quad (3-164)$$

The mode mixture coefficients are defined by

$$\begin{aligned} \beta &= \frac{\Delta u_{II}}{\Delta u_I} \\ \gamma &= \frac{\Delta u_{III}}{\Delta u_I} \end{aligned}$$

where  $\Delta u_I$ ,  $\Delta u_{II}$ , and  $\Delta u_{III}$  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 3-164 simplifies to

$$K_I = \sqrt{\frac{E_{\text{eff}}}{1 + \beta^2}} J$$

$$K_{II} = \sqrt{\frac{E_{\text{eff}}}{1 + \frac{1}{\beta^2}}} J$$

The definition of the effective Young's modulus depends on the stress state at the crack tip

$$E_{\text{eff}} = \begin{cases} \frac{E}{1 - \nu^2} & \text{if plane strain} \\ E & \text{if plane stress} \end{cases}$$

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.

# 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, 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, Beam, or Membrane interface. Mathematically, the mapping of a generic expression  $E$  from a source domain  $\Omega_{\text{src}}$  to a point  $\mathbf{X}_{\text{dst}}$  on the destination boundary (or edge)  $\Gamma_{\text{dst}}$  can be described by

$$E_{\text{src}} = \text{map}[E(\mathbf{X}_{\text{src}}), \mathbf{X}_{\text{dst}}] \text{ on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}} \quad (3-165)$$

The mapping is such that for each point  $\mathbf{X}_{\text{dst}}$ , it finds the closet point  $\mathbf{X}_{\text{src}}$  in  $\Omega_{\text{src}}$  at which to evaluate  $E$ . From [Equation 3-165](#) 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  $\Gamma_{\text{dst}}$  that lies within  $\Omega_{\text{src}}$ .

With the definition in [Equation 3-165](#), the rigid connection type is described by adding a constraint on  $\Gamma_{\text{dst}}$  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

$$\Delta\mathbf{u} = \text{map}[\mathbf{u}(\mathbf{X}_{\text{src}}), \mathbf{X}_{\text{dst}}] - \mathbf{u}(\mathbf{X}_{\text{dst}}) \text{ on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

From the relative displacement vector  $\Delta\mathbf{u}$ , the spring force  $\mathbf{f}$  of the penalty formulation is defined by Hooke's law

$$\mathbf{f} = K\Delta\mathbf{u} \text{ on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

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,  $\Gamma_{\text{dst}}$ , such that the spring force  $\mathbf{f}_1$  in local coordinates is defined by

$$\mathbf{f}_1 = K_1(T \cdot \Delta\mathbf{u}) = K_1\Delta\mathbf{u}_1$$

where  $T$  is the transformation matrix from local to global coordinates and  $K_1$  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 we get

$$\Delta\mathbf{u}_{1,e} = \Delta\mathbf{u}_1 - \mathbf{u}_{1,p} \text{ on } \Gamma_{\text{dst}} \cap \Omega_{\text{src}}$$

where  $\mathbf{u}_{1,e}$  is the elastic displacement, and  $\mathbf{u}_{1,p}$  is the plastic displacement (or slip) on the destination. The definitions of the spring force and the weak contribution are unaffected.

The slip vector  $\mathbf{u}_{1,p}$  is defined through a local constitutive model based on the flow theory of plasticity. When the destination physics is either a Truss 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,  $\mathbf{u}_{1,p} = \{0, 0, u_{pn}\}$ . The constitutive model is then summarized by

$$\begin{aligned} F &\equiv |f_n| - c \leq 0 \\ \dot{u}_{pn} &= \lambda \frac{\partial F}{\partial f_n} \\ \dot{u}_{pe} &= \lambda \\ F \leq 0, \lambda &\geq 0, \lambda F = 0 \end{aligned} \tag{3-166}$$



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  $\lambda$ , and the accumulated slip  $u_{pe}$ . The last row of Equation 3-166 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,  $\mathbf{u}_{1,p} = \{u_{t1}, u_{t2}, 0\}$ . The constitutive model then takes a slightly different form, and it can be summarized by

$$\begin{aligned}
 F &\equiv \|\mathbf{f}_t\| - c \leq 0 \\
 \left\{ \dot{u}_{t1,p}, \dot{u}_{t2,p}, 0 \right\} &= \lambda \frac{\partial F}{\partial \mathbf{f}_t} \\
 \dot{u}_{pe} &= \lambda \\
 F \leq 0, \lambda &\geq 0, \lambda F = 0
 \end{aligned} \tag{3-167}$$

where  $\mathbf{f}_t$  is a vector containing the two tangent components of the local spring force  $\mathbf{f}_1$ .

The rate equations in the constitutive models given by Equation 3-166 and Equation 3-167 are implemented through a backward Euler discretization. The resulting set of nonlinear algebraic equations is solved using Newton's method to find  $\lambda$ ,  $u_{pe}$ , and  $\mathbf{u}_{1,p}$ . Optionally, it is also possible to compute the dissipated energy due to friction,  $W_p$ , by also solving the following rate equation

$$\dot{W}_p = \lambda \frac{\partial F}{\partial \mathbf{f}_1} \cdot \mathbf{f}_1$$



---

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  $\theta_{xl}$  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 non-convergence of the solver.

---

# 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  $\mathbf{u}_p$  and a rotation  $\Theta_p$ , then the displacement of any other point  $i$  can under rigid body assumptions be computed as

$$\tilde{\mathbf{u}}_i = \mathbf{u}_p + \Theta_p \times \mathbf{r}_i$$

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  $\mathbf{R}$ :

$$\tilde{\mathbf{u}}_i = \mathbf{u}_p + (\mathbf{R}(\Theta_p) - \mathbf{I})\mathbf{r}_i$$

Here,  $\mathbf{I}$  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.

$$\mathbf{R} = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix}$$

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  $\phi$ :

$$\mathbf{R} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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  $\Theta_p$  are replaced with the sought-after average displacement  $\mathbf{u}_{\text{avg}}$  and average rotation  $\Theta_{\text{avg}}$  respectively. The vector  $\Theta_{\text{avg}}$  is the axis-angle representation of the rotation. The magnitude,  $\|\Theta_{\text{avg}}\|$ , 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.

$$\frac{\partial}{\partial t} \mathbf{u}_{\text{avg}} = \dot{\mathbf{u}}_{\text{avg}}; \quad \frac{\partial}{\partial t} \dot{\mathbf{u}}_{\text{avg}} = \ddot{\mathbf{u}}_{\text{avg}}$$

Analogous, the average angular velocity and accelerations are defined as

$$\frac{\partial}{\partial t} \Theta_{\text{avg}} = \dot{\Theta}_{\text{avg}}; \quad \frac{\partial}{\partial t} \dot{\Theta}_{\text{avg}} = \ddot{\Theta}_{\text{avg}}$$

# 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

$$W_s = \int_0^{\varepsilon_{el}} \sigma : d\varepsilon_{el} \quad (3-168)$$

If the linear elasticity is assumed, then

$$\sigma = \sigma_i + C : \varepsilon_{el}$$

where  $\sigma_i$  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

$$W_h = \frac{1}{4} \text{real}(\sigma : \bar{\varepsilon})$$

where the stress and strain are considered as complex quantities, and the overline denotes a complex conjugate.

### **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 3-168](#) 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  $\rho$  is the mass density, and  $v$  is the velocity.

In Frequency Domain, the kinetic energy is defined as

$$W_k = \frac{1}{4}\rho\omega^2|v|^2$$

which represents the cycle average.

### *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

$$Q_h = \sigma:\dot{\epsilon}$$

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

$$Q_h = \frac{1}{2} \text{real}(\sigma : \overline{i\omega \epsilon})$$

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

$$\mathbf{I} = -\boldsymbol{\sigma} \cdot \mathbf{v}$$

The reason for the minus sign in the definition is that if you put a pressure on an external 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:

$$\mathbf{I} = -\boldsymbol{\sigma} \cdot \tilde{\mathbf{v}}$$

The *mechanical energy flux* is in the frequency domain defined as a real quantity, the cycle average of the complex mechanical energy flux.

$$\mathbf{I} = \frac{1}{2} \text{real}(-\boldsymbol{\sigma} \cdot \tilde{\mathbf{v}})$$



## Energy Variables

The energy variables used in the Structural Mechanics interfaces are summarized in [Table 3-8](#).

TABLE 3-8: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE <sup>1</sup>	COMMENT
phys.Ws	Elastic strain energy density	J/m <sup>3</sup>	solid, mbd, shell, plate, mbrn, beam, truss	
phys.WsM	Membrane strain energy density	J/m <sup>2</sup>	shell, plate	Contributes to phys.Ws
phys.WsB	Bending strain energy	J/m <sup>2</sup>	shell, plate	Contributes to phys.Ws
phys.WsS	Shear strain energy	J/m <sup>2</sup>	shell, plate	Contributes to phys.Ws
phys.WsM	Membrane strain energy density	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	Global variable containing integration and summation of all phys.Ws contributions in a physics interface.
phys.Wk	Kinetic energy density	J/m <sup>3</sup>	solid, mbd, shell, plate, mbrn, beam, truss	

TABLE 3-8: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE <sup>1</sup>	COMMENT
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 <sup>3</sup>	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 <sup>3</sup>	solid	Energy density dissipated by creep.
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 <sup>3</sup>	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.Wed	Dielectric dissipation density	J/m <sup>3</sup>	solid	Piezoelectric electrical damping

TABLE 3-8: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE <sup>1</sup>	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 <sup>3</sup>	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.l*	Mechanical energy flux, *coordinate	W/m <sup>2</sup>	solid, mbd	
phys.lcomplex*	Complex mechanical energy flux, *coordinate	W/m <sup>2</sup>	solid, mbd	Frequency domain only

<sup>1</sup>The availability of physics interfaces is based on the license for a module.

# References

## REFERENCES FOR NONLINEAR STRUCTURAL MODULE MATERIALS

1. G.A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, John Wiley & Sons, 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. S. Hartmann and P. Neff, “Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility,” *International Journal of Solids and Structures*, vol 40, pp. 2767-2791, 2003.
11. M. Attard, “Finite Strain - Isotropic Hyperelasticity,” *Int. J. Solids Struct.*, vol. 40, pp. 4353–4378, 2003.
12. 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.
13. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, 2009.
14. A. Gent, “A new Constitutive Relation for Rubbers,” *Rubber Chem. Technol.*, vol. 69, pp. 59–61, 1996.

15. C. Horgan and G. Saccomandi, "Constitutive Models for Compressible Nonlinearly Elastic Materials with Limiting Chain Extensibility," *J. Elasticity*, vol. 7, pp. 123–138, 2004.
16. B. Storakers, "On Material Representation and Constitutive Branching in Finite Compressible Elasticity," *J. Mech. Phys. Solids*, vol. 34, pp. 125–145, 1986.
17. Y. Gao, "Large Deformation Field Near a Crack Tip in a Rubber-like Material," *Theor. Appl. Fract. Mec.*, vol. 26, pp. 155–162, 1997.
18. R.L. Taylor, "Thermomechanical analysis of viscoelastic solids," *Int. J. Numer. Meth.*, vol. 2, pp. 45–59, 1970.
19. 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.
20. 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.
21. 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.
22. J. Bergstrom. *Mechanics of Solid Polymers: Theory and Computational Modeling*. Elsevier 2015.
23. A. Souza et al., "Three-dimensional model for solids undergoing stress-induced phase transformations," *European J. Mech. A Solids*, vol. 17, pp. 789–806, 1988.
24. 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.
25. D. Lagoudas (Ed.) *Shape Memory Alloys: Modeling and Engineering Applications*, Springer, 2008.
26. D. Lagoudas et al, "Constitutive model for the numerical analysis of phase transformation in polycrystalline shape memory alloys," *Int. J. Plast.*, vol. 32–33, pp. 155–183. 2012.
27. L. Mullins, "Effect of stretching on the properties of rubber," *Journal of Rubber Research*, vol. 16, pp. 275–289, 1947.

28. 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.
29. L. Mullins. “Softening of rubber by deformation,” *Rubber Chem. Technol.*, vol. 42, pp. 339–362, 1969.
30. 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.
31. 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.
32. Z. Guo and L. Sluys, “Computational modelling of the stress-softening phenomenon of rubber-like materials under cyclic loading,” *European Journal of Mechanics A/Solids*, vol. 25, pp. 877–896, 2006.
33. C. Miehe, “Discontinuous and continuous damage evolution in Ogden-type large strain elastic materials,” *European Journal of Mechanics, A/Solids*, vol. 14, pp. 697–720, 1995.
34. R. Koeller, “Application of fractional calculus to the theory of viscoelasticity,” *J. Appl. Mech.*, vol. 51, pp. 299–307, 1984.

#### REFERENCES FOR THE GEOMECHANICS MODULE MATERIALS

1. W.F. Chen and E. Mizuno, *Nonlinear Analysis in Soil Mechanics: Theory and Implementation (Developments in Geotechnical Engineering)*, 3rd ed., Elsevier Science, 1990.
2. D. Drucker and W. Prager, “Soil Mechanics and Plastic Analysis or Limit Design,” *Quarterly of Applied Mathematics*, vol. 10, no. 2, pp. 157–165. 1952.
3. S. Dolarevic and A. Ibrahimbegovic, “A modified three-surface elasto-plastic cap model and its numerical implementation,” *Computers and Structures*, vol. 85, pp. 419-430, 2007.
4. B. Bresler and K. Pister, “Strength of Concrete Under Combined Stresses,” *ACI Journal*, vol. 551, no. 9, pp. 321–345, 1958.
5. H. Matsuoka and T. Nakai, “Stress-deformation and Strength Characteristics of Soil Under Three Different Principal Stresses,” *Proc. JSCE*, vol. 232, 1974.

6. H. Matsuoka and T. Nakai, "Relationship Among Tresca, Mises, Mohr-Coulomb, and Matsuoka-Nakai Failure Criteria," *Soils and Foundations*, vol. 25, no. 4, pp. 123–128, 1985.
7. H.S. Yu, *Plasticity and Geotechnics*, Springer, 2006.
8. V. Marinou, P. Marinou, and E. Hoek, "The Geological Strength Index: Applications and Limitations," *Bull. Eng. Geol. Environ.*, vol. 64, pp. 55–65, 2005.
9. J. Jaeger, N. G. Cook, and R. Zimmerman, *Fundamentals of Rock Mechanics*, 4th ed., Wiley-Blackwell, 2007.
10. G. C. Nayak and O. C. Zienkiewicz, "Convenient Form of Stress Invariants for Plasticity," *J. Struct. Div. ASCE*, vol. 98, pp. 949–954, 1972.
11. A.J. Abbo and S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr-Coulomb Yield Criterion," *Computers and Structures*, vol. 54, no. 3, pp. 427–441, 1995.
12. 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.
13. B.H.G. Brady and E.T. Brown, *Rock Mechanics for Underground Mining*, 3rd ed., Springer, 2004.
14. H.A. Taiebat and J.P. Carter, "Flow Rule Effects in the Tresca Model," *Computer and Geotechnics*, vol. 35, pp. 500–503, 2008.
15. 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.
16. D.M. Wood, *Soil Behaviour and Critical State Soil Mechanics*, Cambridge University Press, 2007.
17. D.M. Potts and L. Zdravkovic, *Finite Element Analysis in Geotechnical Engineering*, Thomas Telford, 1999.
18. W. Tiejing and others, *Stress-strain Relation for Concrete Under Triaxial Loading*, 16th ASCE Engineering Mechanics Conference, 2003.
19. W.F. Chen, *Plasticity in Reinforced Concrete*, McGraw-Hill, 1982.
20. N. Ottosen, "A Failure Criterion for Concrete," *J. Eng. Mech. Division, ASCE*, vol. 103, no. 4, pp. 527–535, 1977.

21. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modelling*, Elsevier, 2005.
22. J. Suebsuk, S. Horpibulsuk, and M. Liu, “Modified Structured Cam Clay: A generalised critical state model for destructured, naturally structured and artificially structured clays,” *Computers and Geotechnics*, vol. 37, pp. 956–968, 2010.
23. M. Lui and J.P. Carter, “A structured Cam Clay model,” *Can. Geotech. J.*, vol. 39, pp. 1313–1332, 2002.
24. E. Alonso, A. Gens, and A. Josa, “A constitutive model for partially saturated soils,” *Géotechnique*, vol. 40, 1990.
25. D. Pedroso and M. Farias, “Extended Barcelona Basic Model for unsaturated soil under cyclic loadings,” *Computers and Geotechnics*, vol. 38, no. 5, pp. 731–740, 2011.
26. T. Bower, *Constitutive modelling of soils and fibre-reinforced soils*, PhD Thesis, Cardiff University 2017.

#### REFERENCES FOR THE NONLINEAR ELASTIC MATERIAL THEORY

1. W. Ramberg and W.R. Osgood, “Description of stress-strain curves by three parameters,” *NACA Technical Note*, no. 902, 1943.
2. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, 2009.
3. B.O. Hardin and V.P. Drnevich, “Shear modulus and damping in soils: Design equations and curves,” *J. Soil Mechanics and Foundations Division*, ASCE 98, pp. 667–692, 1972.
4. J.M. Duncan and C.Y. Chang, “Nonlinear analysis of stress and strain in soils,” *J. Soil Mechanics and Foundations Division*, vol. 96, no. 5, pp. 1629–1653, 1970.
5. 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.
6. 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

1. 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.* 24 (1) pp. 1–26, 2007.
2. M. Jirasek and Z. Bazant, *Inelastic Analysis of Structures, Ch.20 General Elastoplastic Constitutive Models*. Wiley, 2001.
3. J. Simo and T. Hughes, *Computational Inelasticity*, Springer, 1998.
4. J. Simo, "Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 99, pp. 61–112, 1992.
5. J. Lubliner, *Plasticity Theory*, Dover, 2008.
6. R. Hill, "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proc. Roy. Soc. London*, vol. 193, pp. 281–297, 1948.
7. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modeling*, Elsevier Science, 2005.
8. S. Shima and M. Oyane. "Plasticity theory for porous metals," *International Journal of Mechanical Sciences*, vol. 18, pp. 285–291, 1976.
9. A. Gurson, "Continuum theory of ductile rupture by void nucleation and growth: Part I – Yield criteria and flow rules for porous ductile media," *Journal of Engineering Materials and Technology*, vol. 99, pp. 2–15, 1977.
10. V. Tvergaard and A. Needleman, "Analysis of the cup-cone fracture in a round tensile bar," *Acta Metallurgica*, vol. 32, pp. 157–169, 1984.
11. A. Needleman and V. Tvergaard. "An analysis of ductile rupture in notched bars," *Journal of the Mechanics and Physics of Solids*, vol. 32, pp. 461–490, 1984.
12. K. Nahshon and Z. Xue, "A modified Gurson model and its application to punch-out experiments," *Engineering Fracture Mechanics*, vol. 76, pp. 997–1009, 2009.
13. N. Fleck, L. Kuhn, and R. McMeeking, "Yielding of metal powder bonded by isolated contacts," *Journal of the Mechanics and Physics of Solids*, vol. 40, pp. 1139–1162, 1992.
14. P. Redanz, "Numerical modelling of the powder compaction of a cup," *European Journal of Mechanics - A/Solids*, vol. 18, pp.399–413, 1999.

15. J. Cedergren, N. Sorensen, and A. Bergmark, “Three-dimensional analysis of compaction of metal powder,” *Mechanics of Materials*, vol. 34, pp. 43–59, 2004.
16. S. Tsai and E. Wu, “A general theory of strength for anisotropic materials,” *Journal of Composite Materials*, vol. 5, pp. 58-80, 1971.
17. P. Hopkins, “Benchmarks for Membrane and Bending Analysis of Laminated Shells. Part 2: Strength Analysis,” NAFEMS Ltd, 2005.
18. Z. Hashin, “Failure Criteria for Unidirectional Fiber Composites”, *Journal of Applied Mechanics*, 47, 329,1980.
19. 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”, *Structural and Multidisciplinary Optimization*, 47, 735–747, 2013.
20. C. G. Davila, P. P. Camanho, and C. A. Rose. “Failure Criteria for FRP Laminates,” *Journal of Composites Materials*, 39, 2005.

#### REFERENCES FOR THE DAMAGE MODELS

1. R. de Borst, M. Crisfield, J. Remmers, and C. Verhoosel, *Non-linear Finite Element Analyses of Solids and Structures*, John Wiley & Sons, 2012.
2. M. Jirásek, “Damage and smeared crack models,” pp. 1–49 in *Numerical Modeling of Concrete Cracking*, G. Hofstetter and G. Meschke eds., Springer, 2011.
3. J. Mazars, “A description of micro and macroscale damage of concrete structures,” *Engineering Fracture Mechanics*, vol. 25, pp. 729–737, 1986.
4. Z. Bazant and B. Oh, “Crack band theory for fracture in concrete,” *Materials and Structures*, vol. 16, pp. 155–177, 1983.
5. M. Jirásek and M. Bauer, “Numerical aspects of the crack band approach,” *Computers and Structures*, vol. 110, pp. 60-78, 2012.
6. R. Peerlings and others, “Gradient enhanced damage for quasi-brittle materials,” *Int. J. Numer. Meth. Eng.*, vol. 39, pp. 3391–3403, 1996.
7. 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.
8. C. Miehe, L.S. Schänzel, and H. Ulmer, “Phase field modeling of fracture in multi-physics system. 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, 2014.

9. 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 THE PIEZOELECTRICITY MULTIPHYSICS INTERFACE

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*, IEEE International, pp. 162–169, 2002.

7. C.A. Balanis, “Electrical Properties of Matter,” *Advanced Engineering Electromagnetics*, John Wiley & Sons, 1989.

8. J. Yang, *An Introduction to the Theory of Piezoelectricity*, Springer Science and Business Media, N.Y., 2005.

#### REFERENCES FOR MAGNETOSTRICTION

1. Chikazumi, *Physics of Ferromagnetism*, Oxford University Press, New York, 1997.

2. 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.

3. X.E. Liu and X.J. Zheng, “A Nonlinear Constitutive Model for Magnetostrictive Materials”, *Acta Mech. Sinica*, vol. 21, pp. 278–285, 2005.

4. D.C. Jiles, *Introduction to Magnetism and Magnetic Materials*, 2nd ed., Chapman & Hall, London, 1998.
5. M.J. Dapino, "Nonlinear and Hysteretic Magnetomechanical Model for Magnetostrictive Transducers," PhD Dissertation, Iowa State University, Ames, Iowa, 1999.
6. 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 BOUNDARY CONDITIONS

1. 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), Nort-Holland, 1983.
2. B. Lallane and M. Touratier, "Aeroelastic Vibrations and Stability in Cyclic Symmetric Domains," *International Journal of Rotating Machinery*, vol. 6, no. 6, pp. 445–452, 2000.
3. Y.C. Fung, *Foundations of Solid Mechanics*, Prentice-Hall, 1965, page 99.



#### REFERENCES FOR CONTACT ANALYSIS AND DECOHESION

1. L. De Lorenzis, D. Fernando, and J.G. Teng, "Coupled mixed-mode cohesive zone modeling of interfacial debonding in plated beams," *International Journal of Solids and Structures*, vol. 50, pp. 2477–2494, 2013.
2. 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," *International Journal of Solids and Structures*, vol. 45, no. 5, pp. 1497–1512, 2008.
3. 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.
4. N. Valoroso and L. Champeny, "A damage-mechanics-based approach for modelling decohesion in adhesively bonded assemblies," *Engineering Fracture Mechanics*, vol. 73, pp. 2774–2801, 2006.
5. L. Allix and A Corigliano, "Geometrical and interfacial non-linearities in the analysis of delamination in composites," *International Journal of Solids and Structures*, vol. 36, pp. 2189–2216, 1999

# 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, or axisymmetric bodies. In 2D, 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 <https://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 boundary condition where boundaries are free, with no loads or constraints), and **Initial Values**. 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`.

## 2D APPROXIMATION

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** check box 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 and Circumferential Modes](#) in the *Structural Mechanics Theory* chapter.

## AXIAL SYMMETRY APPROXIMATION



Select **Circumferential mode extension (time-harmonic)** to prescribe a circumferential wave number to be used in eigenfrequency or frequency domain studies. When selected, enter the **Circumferential mode number  $m$** .

For more information, see [Out-of-Plane and Circumferential Modes](#) in the *Structural Mechanics Theory* chapter.



*Eigenfrequency Analysis of a Free Cylinder*: Application Library path **Structural\_Mechanics\_Module/Verification\_Examples/free\_cylinder**.

## THICKNESS



For 2D components, enter a value or expression for the **Thickness  $d$** . The default value of 1 m is suitable for plane strain models, where it represents a unit-depth slice, for example. For plane stress models, enter the actual thickness, which should be small compared to the size of the plate for the plane stress assumption to be valid.

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 which do not have an explicit thickness property will implicitly assume unit thickness.

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** 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.

#### 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** check box to activate this functionality.

Enter the **Maximum frequency to resolve** in the model,  $f_{\max, \text{sol}}$ .

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 set up 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.

#### REFERENCE POINT FOR MOMENT COMPUTATION

Enter the coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$  (variable `refpnt`). The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

#### TYPICAL WAVE SPEED FOR PERFECTLY MATCHED LAYERS

The typical wave speed  $c_{\text{ref}}$  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 expression in the **Typical wave speed for perfectly matched layers** field.

Note: This section is only available with COMSOL products that include PMLs (see <https://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 nodes. This means that it can, and often does, connect elements from different physics interfaces.

In the drop-down 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

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.

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.

## 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 name in the 2D cases as a parameter or variable name, since it is still 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 which 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.



---

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.

---



- [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 Crankshaft*: Application Library path **COMSOL\_Multiphysics/Structural\_Mechanics/crankshaft**
- 

### *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 SUBMENU

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**, and **Domain Constraints**.

The submenus at the **Boundary** level are **Connections**, **Pairs**, **Mass**, **Spring**, and **Damper**, and **More Constraints**.

There are also the **Edges** and **Points** submenus.

Note: Some submenus are only present with certain COMSOL products.

#### FORCE LOADS

Note that you can add force loads acting on all levels of the geometry for the physics interface. Add a:

- **Body Load** to domains (to model gravity effects, for example).
- **Boundary Load** to boundaries (a pressure acting on a boundary, for example).
- **Edge Load** to edges in 3D (a force distributed along an edge, for example).
- **Point Load** to points (concentrated forces at points).



If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence.



For 2D axisymmetric components, COMSOL Multiphysics takes the axial symmetry boundaries (at  $r = 0$ ) into account and automatically adds an **Axial Symmetry** node to the component that is valid on the axial symmetry boundaries only.

- Added Mass
- Antisymmetry
- Attachment
- Beam Connection
- Body Load
- Bolt Pretension
- Bolt Selection
- Bolt Thread Contact
- Boundary Load
- Boundary Pair
- Cell Periodicity
- Change Thickness
- Contact
- Crack
- Edge Load
- Elastic Predeformation
- Elastoplastic Soil Material
- External Strain
- External Stress
- External Stress-Strain Relation
- Fixed Constraint
- Free
- Gravity
- Hyperelastic Material
- Initial Values
- Linear Elastic Material
- Low-Reflecting Boundary
- Magnetostrictive Material
- Nonlinear Elastic Material
- Periodic Condition
- Piezoelectric Material
- Point Load
- Point Load (on Axis)
- Port
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Rigid Motion Suppression
- Rigid Connector
- Rigid Domain
- Ring Load
- Roller
- Rotating Frame
- Shape Memory Alloy
- Shell Connection
- Spring Foundation
- Spring-Damper
- Stress Linearization
- Symmetry
- Thin Elastic Layer
- Thin-Film Damping
- Thread Boundary Selection
- Wave Speeds



In the *COMSOL Multiphysics Reference Manual*, 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.

---

These subnodes are available from main parent nodes as indicated in the documentation:

- Activation
- Adhesion
- Applied Force (Rigid Connector)
- Applied Force (Rigid Domain)
- Applied Moment (Rigid Connector)
- Applied Moment (Rigid Domain)
- Center of Mass Nodes (Boundary, Edge, Point)
- Center of Rotation Nodes (Boundary, Edge, Point)
- Conduction Loss (Time-Harmonic)
- Concrete
- Creep
- Damage
- Damping
- Decohesion
- Destination Point (for Spring-Damper)
- Dielectric Loss
- Elastic Predeformation
- External Stress
- Face Load
- Fixed Constraint (Rigid Domain)
- Friction
- Hygroscopic Swelling
- Initial Stress and Strain
- Initial Values (Rigid Domain)
- J-Integral
- Location Nodes (Boundary, Edge, Point)
- Mass and Moment of Inertia (Rigid Connector)
- Mass and Moment of Inertia (Rigid Domain)
- Mechanical Damping
- Phase
- Plasticity
- Porous Plasticity
- Prescribed Displacement/Rotation
- Predeformation
- Rocks
- Safety
- Set Variables
- Slip Velocity
- Soil Plasticity
- Source Point (for Spring-Damper)
- Spring Foundation (Rigid Connector)
- Spring Foundation (Rigid Domain)
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Attachment)
- Thermal Expansion (Rigid Connector)
- 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 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  $\mathbf{u}$**  (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>Initial Values**

**Layered Shell>Initial Values**

**Membrane>Initial Values**

**Truss>Initial Values**

#### *Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>Solid Mechanics>Initial Values**

Physics tab with **Layered Shell** selected:

**Boundaries>Layered Shell>Initial Values**

Physics tab with **Membrane** selected:

**Boundaries>Membrane>Initial Values**

Physics tab with **Truss** selected:

**Edges>Truss>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

### **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>Change Thickness**

**Multibody Dynamics>Change Thickness**

#### *Ribbon*

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

**Domains>Solid Mechanics>Change Thickness**

**Domains>Multibody Dynamics>Change Thickness**

## *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:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Thermal Expansion (for Materials)</li><li>• Hygroscopic Swelling</li><li>• Initial Stress and Strain</li><li>• External Stress</li><li>• External Strain</li><li>• Damping</li><li>• Viscoelasticity</li><li>• Plasticity</li><li>• Creep</li></ul> | <ul style="list-style-type: none"><li>• Viscoplasticity</li><li>• Porous Plasticity</li><li>• Soil Plasticity</li><li>• Concrete</li><li>• Rocks</li><li>• Damage</li><li>• Activation</li><li>• Safety</li></ul> |
|---|---|

Note: Some options are only available with certain COMSOL products (see <https://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 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.



This section is not present in the in the Layered Shell interface.

## LINEAR ELASTIC MATERIAL

Define the **Solid model** and the linear elastic material properties.

### *Solid Model*

Select a linear elastic **Solid model**: **Isotropic**, **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, so that its stiffness depends on the properties  $E_i$ ,  $\nu_{ij}$ , and  $G_{ij}$ .
- **Anisotropic** for a linear elastic material that has different material properties in different directions, and the stiffness comes from the symmetric elasticity matrix,  $D$ .

Note: The **Orthotropic** and **Anisotropic** options are only available with certain COMSOL products (see <https://www.comsol.com/products/specifications/>)



In the Layered Shell interface, the chosen solid model 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](#)

### *Density*

The default **Density**  $\rho$  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  $T_{\text{ref}}$  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 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 Solid model**, 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	$\nu =$	$\nu$	$\frac{E}{2G} - 1$	$\frac{1}{2} \left( 1 - \frac{3G}{3K+G} \right)$	$\frac{\lambda}{2(\lambda + \mu)}$
Bulk modulus	$K =$	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	$K$	$\lambda + \frac{2\mu}{3}$
Shear modulus	$G =$	$\frac{E}{2(1+\nu)}$	$G$	$G$	$\mu$
Lamé parameter $\lambda$	$\lambda =$	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K - \frac{2G}{3}$	$\lambda$

TABLE 4-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Lamé parameter $\mu$	$\mu =$	$\frac{E}{2(1+\nu)}$	$G$	$G$	$\mu$
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**  $\nu$ .
- **Shear modulus**  $G$ .
- **Bulk modulus**  $K$ .
- **Lamé parameter**  $\lambda$  and **Lamé parameter**  $\mu$ .
- **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 **Solid model** list, the material properties vary in orthogonal directions only. 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**  $E$ , **Poisson's ratio**  $\nu$ , and the **Shear modulus**  $G$ . This defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$



$\nu_{ij}$  is defined differently depending on the application field. It is easy to transform among definitions, but check which one the material uses.

### *Specification of Elastic Properties for Anisotropic Materials*

When **Anisotropic** is selected from the **Solid model** list, the material properties vary in all directions, and the stiffness comes from the symmetric **Elasticity matrix, D**. 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, see the [Mixed Formulation](#) section in the *Structural Mechanics Theory* chapter.

From the **Use mixed formulation** list, select **None**, **Pressure formulation**, or **Strain formulation**.

## **GEOMETRIC NONLINEARITY**

The settings in this section affect the behavior of the selected domains in a geometrically nonlinear analysis.

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. Select the **Force linear strains** check box to always use a small strain formulation, irrespective of the setting in the study step.

When a geometrically nonlinear formulation is used, the elastic deformations used for computing the stresses can be obtained in two different ways if inelastic deformations are present: additive decomposition and multiplicative decomposition. The default is to use multiplicative decomposition. Select **Additive strain decomposition** to change to an assumption of additivity.

Note: This section is only available with COMSOL products that support geometrically nonlinear analysis (see <https://www.comsol.com/products/specifications/>).




- 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 in contact analysis, where the study is always geometrically nonlinear, but it is possible that a geometrically linear formulation is sufficient in the material.
- When a multiplicative decomposition is used, the order of the subnodes to **Linear Elastic Material** matters. The inelastic deformations are assumed to have occurred in the same order as the subnodes appear in the model tree.
- In versions prior to 5.3, only the additive strain decomposition method was available. If you want to revert to the previous behavior, select **Additive strain decomposition**. If the results then differ significantly, probably the assumption of additivity is questionable, however.
- In models created in a version prior to 4.2a, a check box named **Include geometric nonlinearity** may be visible in this section. It is displayed only if geometric nonlinearity was originally used for the selected domains. Once the check box is cleared in this **Settings** window, it is permanently removed and the study step assumes control over the selection of geometric nonlinearity.  
When **Include geometric nonlinearity** is selected in this section, it automatically also selects the **Include geometric nonlinearity** check box in the study **Settings** window.



- [Modeling Geometric Nonlinearity](#)
- [Inelastic Strain Contributions](#)
- [Studies and Solvers](#) in the *COMSOL Multiphysics Reference Manual*.

## ENERGY DISSIPATION

You can select to compute and store various energy dissipation variables in a time dependent analysis. Doing so will add extra degrees of freedom to the model.

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by for example creep, plasticity, viscoplasticity, viscoelasticity, or damping.



- [Dissipated Energy](#)
  - [Energy Variables](#)
- 

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Material Models>Linear Elastic Material**

**Layered Shell>Material Models>Linear Elastic Material**

**Multibody Dynamics>Linear Elastic Material**

### *Ribbon*

Physics tab with **Solid Mechanics** 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>Multibody Dynamics>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.



By adding the following subnodes to the **Nonlinear Elastic Material** node you can incorporate many other effects:

<ul style="list-style-type: none"> <li>• Thermal Expansion (for Materials)</li> <li>• Hygroscopic Swelling</li> <li>• Initial Stress and Strain</li> <li>• External Stress</li> <li>• External Strain</li> <li>• Damping</li> <li>• Viscoelasticity</li> <li>• Plasticity</li> </ul>	<ul style="list-style-type: none"> <li>• Creep</li> <li>• Viscoplasticity</li> <li>• Porous Plasticity</li> <li>• Soil Plasticity</li> <li>• Concrete</li> <li>• Rocks</li> <li>• Safety</li> </ul>
--	---

Note: Some options are only available with certain COMSOL products (see <https://www.comsol.com/products/specifications/>)

#### COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used 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**, or **User defined**.

#### *Density*

All nonlinear elastic material models have density as an input. The default **Density**  $\rho$  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  $T_{\text{ref}}$  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 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 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**, or **Strain formulation**.

*Ramberg–Osgood, Power Law, Hyperbolic Law, Hardin–Drnevich, Duncan–Chang, or Duncan–Selig*

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**  $\nu$ .

- **Shear modulus**  $G$ .
- **Bulk modulus**  $K$ .
- For **Ramberg-Osgood**:
  - **Reference stress**  $\sigma_{\text{ref}}$ .
  - **Reference strain**  $\varepsilon_{\text{ref}}$ .
  - **Stress exponent**  $n$ .
- For **Power law** and **Hyperbolic law**:
  - **Reference shear strain**  $\gamma_{\text{ref}}$ .
  - **Strain exponent**  $n$ .
- For **Hardin-Drnevich**, define the **Reference shear strain**  $\gamma_{\text{ref}}$ .
- For **Duncan-Chang**, define the **Ultimate deviatoric stress**  $q_{\text{ult}}$ .
- For **Duncan-Selig**:
  - **Ultimate deviatoric stress**  $q_{\text{ult}}$ .
  - **Ultimate strain**  $\varepsilon_{\text{ult}}$ .

#### *Uniaxial Data*

For **Uniaxial data** the **Uniaxial stress function**  $\sigma_{\text{ax}}$  uses the value **From material** (if it exists) or **User defined**. If **User defined** is selected from the list, the default expression for  $\sigma_{\text{ax}}$  is the linear function  $210[\text{GPa}] * \langle \text{physics} \rangle . \text{eax}$ , which corresponds to a linear elastic material with a Young's modulus of 210 GPa. The variable  $\langle \text{physics} \rangle . \text{eax}$  corresponds to the elastic uniaxial strain in pure axial loading, and is named using the scheme  $\langle \text{physics} \rangle . \text{eax}$ , for example,  $\text{solid} . \text{eax}$ .

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**  $\nu$ .

When you select **Bulk modulus**, the Young's modulus is computed from the tensile part of the **Uniaxial stress function**  $\sigma_{\text{ax}}$ . When you select **Poisson's ratio**, you can either use the tensile part (default), or use the full tensile-compressive function by selecting the check box **Use nonsymmetric stress-strain data**.

### *Shear Data*

For **Shear data** the **Shear stress function**  $\tau$  uses the value **From material** (if it exists) or **User defined**. If **User defined** is selected from the list, the default expression for  $\tau$  is the linear function  $80[\text{GPa}] * \langle \text{physics} \rangle . \text{esh}$ , which corresponds to a linear elastic material with a shear modulus of 80 GPa. The variable  $\langle \text{physics} \rangle . \text{esh}$  corresponds to the elastic shear strain in pure shear loading, and it is named using the scheme  $\langle \text{physics} \rangle . \text{esh}$ , for example,  $\text{solid.esh}$ .

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**  $G$ .

### *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  $(-160[\text{GPa}]) * \text{solid.eelvol}$ , which corresponds to a linear elastic material with a bulk modulus of 160 GPa.
- **Shear modulus**  $G$ .

## **GEOMETRIC NONLINEARITY**

The settings in this section affect the behavior of the selected domains in a geometrically nonlinear analysis.

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. Select the **Force linear strains** check box to always use a small strain formulation, irrespective of the setting in the study step.

When a geometrically nonlinear formulation is used, the elastic deformations used for computing the stresses can be obtained in two different ways if inelastic deformations are present: additive decomposition and multiplicative decomposition. The default is

to use multiplicative decomposition. Select **Additive strain decomposition** to change to an assumption of additivity.




- 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 in contact analysis, where the study is always geometrically nonlinear, but it is possible that a geometrically linear formulation is sufficient in the material.
- When a multiplicative decomposition is used, the order of the subnodes to **Nonlinear Elastic Material** matters. The inelastic deformations are assumed to have occurred in the same order as the subnodes appear in the model tree.
- In versions prior to 5.3, only the additive strain decomposition method was available. If you want to revert to the previous behavior, select **Additive strain decomposition**. If the results then differ significantly, probably the assumption of additivity is questionable, however.



- [Modeling Geometric Nonlinearity](#)
- [Inelastic Strain Contributions](#)
- [Studies and Solvers](#) 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 box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or **Viscoelasticity**.

## 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 which are nonlinear even at infinitesimal strains. It is available in the Solid Mechanics interface. This material model requires a Geomechanics Module license (see <https://www.comsol.com/products/specifications/>).

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](#)
- [Safety](#)

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.

#### **COORDINATE SYSTEM SELECTION**

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used 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**, or **Hardening Soil**.

### Density

All elastoplastic soil models have density as an input. The default **Density**  $\rho$  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  $T_{\text{ref}}$  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 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*.

### Modified Cam-Clay

The **Modified Cam-Clay** options adds the equations and interface for defining the material properties for the modified Cam-Clay soil model.



See also [The Modified Cam-Clay Soil Model](#) in the *Structural Mechanics Theory* chapter.

From the **Specify** list, define the elastic properties either in terms of **Poisson's ratio** or **Shear modulus**.

The defaults for the **Poisson's ratio**  $\nu$  or the **Shear modulus**  $G$ , **Density**  $\rho$ , **Slope of critical state line**  $M$ , **Swelling index**  $\kappa$ , **Compression index**  $\lambda$ , **Initial void ratio**  $e_0$ , and **Void ratio at reference pressure**  $e_{\text{ref}}$  are taken **From material**. For **User defined** enter other values or expressions

Enter a value or expression for the **Reference pressure**  $p_{\text{ref}}$ , and the **Initial consolidation pressure**  $p_{c0}$ .



For the **Slope of critical state line** you can alternatively select **Match to Mohr-Coulomb criterion**, which then matches the slope of the virgin consolidation line to the angle of internal friction. Then select the **Angle of internal friction**  $\phi$  as **From material** or **User defined**.

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 initial consolidation pressure, and the swelling and compression indexes. Then select the **Void ratio at reference pressure**  $e_{\text{ref}}$  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, define the elastic properties either in terms of **Poisson's ratio** or **Shear modulus**.

The defaults for the **Poisson's ratio**  $\nu$  or the **Shear modulus**  $G$ , **Density**  $\rho$ , **Slope of critical state line**  $M$ , **Swelling index for structured clay**  $\kappa_s$ , **Compression index for destructured clay**  $\lambda_d$ , **Angle of internal friction**  $\phi$ , **Initial structure strength**  $p_{bi}$ , **Destructuring index for volumetric deformation**  $d_v$ , **Destructuring index for shear deformation**  $d_s$ , **Plastic potential shape parameter**  $\xi$ , **Initial void ratio**  $e_0$ , **Void ratio at reference pressure for destructured clay**  $e_{\text{ref}d}$ , **Additional void ratio at initial yielding**  $\Delta e_i$ , and **Critical equivalent deviatoric plastic strain**  $e_{dc}^p$  are taken **From material**. For **User defined** enter other values or expressions.



Enter a value or expression for the **Reference pressure**  $p_{\text{ref}}$ , and the **Initial consolidation pressure**  $p_{c0}$ .



For the **Slope of critical state line** you can alternatively select **Match to Mohr-Coulomb criterion**, which then matches the slope of the virgin consolidation line to the angle of internal friction. Then select the **Angle of internal friction**  $\phi$  as **From material** or **User defined**.

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 initial consolidation pressure, and the swelling and compression indexes. Then select the **Void ratio at reference pressure for destructured clay**  $e_{\text{refd}}$  as **From material** or **User defined**. See also [The Modified Structured Cam-Clay Soil Model](#) in the *Structural Mechanics Theory* chapter.

#### *Extended Barcelona Basic*

The defaults for the **Poisson's ratio**  $\nu$  or the **Shear modulus**  $G$ , **Density**  $\rho$ , **Swelling index**  $\kappa$ , **Swelling index for changes in suction**  $\kappa_s$ , **Compression index at saturation**  $\lambda_0$ , **Compression index for changes in suction**  $\lambda_s$ , **Angle of internal friction**  $\phi$ , **Weight parameter**  $w$ , **Soil stiffness parameter**  $m$ , **Plastic potential shape parameter**  $b_s$ , **Tension to suction ratio**  $k$ , **Initial void ratio**  $e_0$ , **Void ratio at reference pressure and saturation**  $e_{\text{ref0}}$ , and **Initial yield value for suction**  $s_{y0}$  are taken **From material**. For **User defined** enter other values or expressions.

Enter a value or expression for the **Initial suction**  $s_0$ , **Suction**  $s$ , the **Reference pressure**  $p_{\text{ref}}$ , and the **Initial consolidation 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 initial consolidation pressure, and the swelling and compression indexes. Then select the **Void ratio at reference pressure and saturation**  $e_{\text{ref0}}$  as **From material** or **User defined**. See also [The Extended Barcelona Basic Soil Model](#) in the *Structural Mechanics Theory* chapter.

### Hardening Soil

The defaults for the **Reference stiffness for primary loading**  $E_{50}^{\text{ref}}$ , **Reference stiffness for unloading and reloading**  $E_{\text{ur}}^{\text{ref}}$ , **Elastoplastic compression modulus**  $K_c$ , **Poisson's ratio**  $\nu$ , **Density**  $\rho$ , **Stress exponent**  $m$ , **Cohesion**  $c$ , **Angle of internal friction**  $\phi$ , **Dilatation angle**  $\psi$ , and **Initial void ratio**  $e_0$  are taken **From material**. For **User defined** enter other values or expressions.

Enter a value or expression for the **Failure ratio**  $R_f$ , the **Reference pressure**  $p_{\text{ref}}$ , and the **Initial consolidation pressure**  $p_{c0}$ .

Select the **Include dilatancy cutoff** check box if needed. The defaults for the **Maximum void ratio**  $e_{\text{max}}$  is taken **From material**. For **User defined** enter other value or expression. Enter a value or expression for the **Initial volumetric strain**  $\varepsilon_{\text{vol}0}$ .



See also [The Hardening Soil Model](#) in the *Structural Mechanics Theory* chapter.

---

### GEOMETRIC NONLINEARITY

The settings in this section affect the behavior of the selected domains in a geometrically nonlinear analysis.

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. Select the **Force linear strains** check box to always use a small strain formulation, irrespective of the setting in the study step.

When a geometrically nonlinear formulation is used, the elastic deformations used for computing the stresses can be obtained in two different ways if inelastic deformations are present: additive decomposition and multiplicative decomposition. The default is

to use multiplicative decomposition. Select **Additive strain decomposition** to change to an assumption of additivity.




- 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 in contact analysis, where the study is always geometrically nonlinear, but it is possible that a geometrically linear formulation is sufficient in the material.
- When a multiplicative decomposition is used, the order of the subnodes to **Elastoplastic Soil Material** matters. The inelastic deformations are assumed to have occurred in the same order as the subnodes appear in the model tree.



- [Modeling Geometric Nonlinearity](#)
- [Inelastic Strain Contributions](#)
- [Studies and Solvers](#) 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 box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity**, or **Viscoelasticity**.

## 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, and Membrane interfaces. This material model requires the Nonlinear Structural Materials Module.

When a hyperelastic material is included in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** check box 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:

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• <a href="#">Thermal Expansion (for Materials)</a></li><li>• <a href="#">Hygroscopic Swelling</a></li><li>• <a href="#">External Stress</a></li><li>• <a href="#">External Strain</a></li></ul> | <ul style="list-style-type: none"><li>• <a href="#">Damping</a></li><li>• <a href="#">Viscoelasticity</a></li><li>• <a href="#">Mullins Effect</a></li><li>• <a href="#">Plasticity</a></li></ul> |
|--|---|



See also [Hyperelastic Material Models](#) in the *Structural Mechanics Theory* chapter.

---

## 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.

---

## HYPERELASTIC MATERIAL

Select a hyperelastic **Material model** from the list and then go to the applicable section for more information.

### *Compressibility*

Hyperelastic materials can use a mixed formulation by adding the negative mean pressure as an extra dependent variable, or a weak constrain to enforce the incompressibility condition. Depending on the hyperelastic material model, select from the **Compressibility** list:

- **Compressible material**
- **Nearly incompressible material, quadratic volumetric strain energy**

- **Nearly incompressible material, Hartmann-Neff volumetric strain energy**
- **Incompressible material**

### Density

All hyperelastic material models have density as an input. The default **Density**  $\rho$  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  $T_{\text{ref}}$  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 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*.

- |  |                                 |
|--|---------------------------------|
| • <a href="#">Neo-Hookean</a>                    | • <a href="#">Arruda–Boyce</a>  |
| • <a href="#">St Venant–Kirchhoff</a>            | • <a href="#">Gent</a>          |
| • <a href="#">Mooney–Rivlin, Two Parameters</a>  | • <a href="#">Blatz–Ko</a>      |
| • <a href="#">Mooney–Rivlin, Five Parameters</a> | • <a href="#">Gao</a>           |
| • <a href="#">Mooney–Rivlin, Nine Parameters</a> | • <a href="#">Murnaghan</a>     |
| • <a href="#">Yeoh</a>                           | • <a href="#">Delfino</a>       |
| • <a href="#">Ogden</a>                          | • <a href="#">Fung</a>          |
| • <a href="#">Storakers</a>                      | • <a href="#">Extended tube</a> |
| • <a href="#">Varga</a>                          | • <a href="#">User Defined</a>  |

#### *Neo-Hookean*

The default values for both **Lamé parameter**  $\lambda$  and **Lamé parameter**  $\mu$  use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$  instead of the Lamé parameter  $\lambda$  to define the volumetric strain energy density. If the **Incompressible material** option is selected from the **Compressibility** list, enter the Lamé parameter  $\mu$  only.

#### *St Venant–Kirchhoff*

The default values for both **Lamé parameter**  $\lambda$  and **Lamé parameter**  $\mu$  use values **From material**.

#### *Mooney–Rivlin, Two Parameters*

For **Mooney–Rivlin, two-parameters** the **Model parameters**  $C_{10}$  and  $C_{01}$  both use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

#### *Mooney–Rivlin, Five Parameters*

For **Mooney–Rivlin, five-parameters** the **Model parameters**  $C_{10}$ ,  $C_{01}$ ,  $C_{20}$ ,  $C_{02}$ , and  $C_{11}$  all use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

#### *Mooney–Rivlin, Nine Parameters*

For **Mooney–Rivlin, nine-parameters** 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**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

#### *Yeoh*

For **Yeoh** the **Model parameters**  $c_1$ ,  $c_2$ , and  $c_3$  all use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

#### *Ogden*

In the table for the **Ogden parameters**, enter values or expressions in each column: **Shear modulus (Pa)**, and **Alpha parameter**.

If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

#### *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*

For **Varga** the **Model parameters**  $c_1$ ,  $c_2$ , and  $c_3$  all use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### *Arruda–Boyce*

For **Arruda-Boyce** the default values for the **Macroscopic shear modulus**  $\mu_0$  and the **Number of segments**  $N$  use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### *Gent*

For **Gent** the default values for the **Macroscopic shear modulus**  $\mu$  and the model parameter  $J_m$  use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### *Blatz–Ko*

For **Blatz-Ko** the **Shear modulus**  $\mu$  and the **Model parameters**  $\beta$  and  $\phi$  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**  $\lambda$  and  $\mu$  use values **From material**.

### *Delfino*

For **Delfino** the **Model parameters**  $a$  and  $b$  use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### *Fung*

For **Fung** the **Coefficient matrix** **A** and **Fung parameter**  $c$  use values **From material**.

The **Coefficient matrix** **A** provides the anisotropic material properties that vary 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.

If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### Extended tube

For **Extended tube** the **Model parameters**  $G_c$ ,  $G_e$ ,  $\alpha$ , and  $\beta$  use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus**  $\kappa$ .

### User Defined

If **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 negative mean pressure as an extra dependent variable. In this case, select from the **Compressibility** list either **Nearly incompressible material** or **Incompressible material**.

If **Nearly incompressible material** is selected, enter the **Isochoric strain energy density**  $W_{\text{siso}}$  and the **Volumetric strain energy density**  $W_{\text{vol}}$ .


If **Incompressible material** is selected, enter the **Isochoric strain energy density**  $W_{\text{siso}}$  only. An extra weak constrain is added to enforce the incompressibility condition  $J_{\text{el}} = 1$ .



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

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

Select the **Calculate dissipated energy** check box to compute the energy dissipated by **Plasticity**.

## LOCATION IN USER INTERFACE

### Context Menus

**Solid Mechanics>Material Models>Hyperelastic Material**

**Layered Shell>Material Models>Hyperelastic Material**

**Shell>Material Models>Layered Hyperelastic Material**



**Membrane>Material Models>Hyperelastic Material**

*Ribbon*

Physics tab with **Solid Mechanics** 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 which are nonlinear even at infinitesimal strains. This material model requires the Nonlinear Structural Materials Module.

By adding the following subnodes to the **Shape Memory Alloy** node you can incorporate other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [External Strain](#)

Note: Some options are only available with certain COMSOL products (see <https://www.comsol.com/products/specifications/>)



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 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 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 any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used 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 **Shape memory alloy reference temperature**  $T_0$ , the **Poisson's ratio**  $\nu$ , and the **Density**  $\rho$  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 parameters**, select which **Transformation parameters** will describe the phase transitions: **Temperature** or **Stress**.

- When **Temperature** is selected from the **Transformation parameters** 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 **Transformation parameters** list, enter the **Martensite start stress**  $\sigma_{Ms}$ , the **Martensite finish stress**  $\sigma_{Mf}$ , the **Slope of martensite limit curve**  $C_M$ , the **Austenite start stress**  $\sigma_{As}$ , the **Austenite finish stress**  $\sigma_{Af}$ , the **Slope of austenite limit curve**  $C_A$ , and the **Measurement temperature**  $T_\sigma$ .

Under the **Maximum transformation strain** list select **Constant** to directly enter the **Maximum transformation strain**  $\epsilon_{tr,max}$ , or **Exponential law** to specify a stress-dependent maximum transformation strain. Under **Exponential law**, enter the **Initial maximum transformation strain**  $\epsilon_{tr,min}$ , the **Ultimate transformation strain**  $\epsilon_{tr,sat}$ , the **Critical stress**  $\sigma_{crit}$ , and the **Saturation exponent**  $k$ . Enter the **Calibration stress level**  $\sigma^*$ .

Under **Phase transformation kinetics**, select the **Transformation function** from the list: **Quadratic**, **Cosine**, **Smooth** or **User defined**.

For **Smooth**, enter the smoothing parameters  $\eta_1$ ,  $\eta_2$ ,  $\eta_3$ , and  $\eta_4$ .

For **User defined** enter the **Yield stress**  $\sigma_{ys}$ , the **Forward transformation law**, and the **Reverse transformation law**.

#### *Souza–Auricchio*

For **Souza–Auricchio** the defaults for the **Poisson's ratio**  $\nu$  and **Density**  $\rho$ , 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 parameters**, enter the **Reference temperature**  $T^*$ , the **Slope of limit curve**  $\beta$ , the **Maximum transformation strain**  $\epsilon_{tr,max}$ , the **Elastic domain radius**  $\sigma_0$ , the **Hardening modulus**  $H_k$ , and the **Indicator function coefficient**  $\gamma$ .

#### *Initial Transformation State*

For Laogudas 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

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are, however, some cases when the use of a small strain formulation for a certain domain is needed. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step. The check box is not selected by default to conserve the properties of the model.

When a geometrically nonlinear formulation is used, the elastic deformations used for computing the stresses can be obtained in two different ways if inelastic deformations are present: additive decomposition and multiplicative decomposition. The default is to use multiplicative decomposition. Select **Additive strain decomposition** to change to an assumption of additivity.

## ENERGY DISSIPATION

You can select to compute and store various energy dissipation variables in a time dependent analysis. Doing so will add extra degrees of freedom to the model.

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated.

## 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**

### *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  $\xi$  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.

### **PHASE TRANSFORMATION DIRECTION**

Select a **Transformation Direction** — **Automatic** or **User Defined**.

### **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:

**Attributes>Phase 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 **Piezoelectric Effect** 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.

This material model requires the Structural Mechanics Module, 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 **Piezoelectric Effect** 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 [Piezoelectric Material](#) in the *Structural Mechanics Theory* chapter.

#### 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 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](#)
- [Default Model Inputs and Model Input](#) in the *COMSOL Multiphysics Reference Manual*.

## GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are, however, some cases when the use of a small strain formulation for a certain domain is needed. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step. The check box is not selected by default to conserve the properties of the model.



- [Modeling Piezoelectric Problems](#)
- [Modeling Geometric Nonlinearity](#)
- [The Electromagnetics Interfaces](#) in the *COMSOL Multiphysics Reference Manual*.

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Material Models>Piezoelectric Material**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>Material Models>Piezoelectric Material**

## *Magnetostrictive Material*

---

The **Magnetostrictive Material** node defines the magnetostrictive material properties.

In case of linear magnetostriction model, 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 [Magnetostriction](#) multiphysics interface together with a **Magnetostriction** multiphysics coupling node and **Ampère's Law, Magnetostrictive** node in the corresponding **Magnetic Fields** interface. **Magnetostrictive Material** node is added by default to the **Solid Mechanics** interface when adding a Magnetostriction multiphysics interface. The interface requires the AC/DC Module and at least one of the following modules: Structural Mechanics, MEMS, Acoustics.



When the **Magnetostrictive Material** node is added to the Structural Mechanics interface in the absence of an active **Magnetostriction** 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 magnetostrictive effect is then *not* included in the corresponding equation system.

By adding the following subnodes to the **Magnetostrictive Material** node you can incorporate other effects:

- [Initial Stress and Strain](#)
- [Thermal Expansion \(for Materials\)](#)
- [Mechanical Damping](#)



See also [Magnetostrictive Material](#) in the *Structural Mechanics Theory* chapter.



## LINEAR MAGNETOSTRICTIVE 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** ( $\mu_{rT}$  or  $\mu_{rS}$ ).
- Select a **Density** ( $\rho$ ).



For entering these matrices, use the following order (Voigt notation), which is the common convention for magnetostrictive 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 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*.

## NONLINEAR MAGNETOSTRICTIVE MATERIAL PROPERTIES

See the corresponding theory section [Nonlinear Magnetostriction](#).

## GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are, however, some cases when the use of a small strain formulation for a certain domain is needed. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step. The check box is not selected by default to conserve the properties of the model.



- [Magnetostriction](#)
- [Modeling Magnetostrictive Materials](#)
- [Modeling Geometric Nonlinearity](#)
- [Ampère's Law, Magnetostrictive](#)
- [The Magnetic Fields Interface in the \*COMSOL Multiphysics Reference Manual\*.](#)

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Material Models>Magnetostrictive Material**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>Material Models>Magnetostrictive Material**

*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**, **Layered Linear Elastic Material**, **Nonlinear Elastic Material**, and **Hyperelastic Material**.



See also [Linear Viscoelastic Materials](#) 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.
- **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Viscoelasticity](#) node in the [Shell and Plate](#) chapter.
- **Layered Linear Elastic Material** in the Membrane interface. See the documentation for the [Viscoelasticity](#) node in the [Membrane](#) chapter.

## 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**  $T_{WLF}$ . The default is 293.15 K.
  - **WLF constant 1**  $C_{1WLF}$ . The default is 17.44.
  - **WLF constant 2**  $C_{2WLF}$ . The default is 51.6 K.
- For **Arrhenius** enter values or expressions for these properties:
  - **Reference temperature**  $T_0$ . The default is 293.15 K.
  - **Activation energy**  $Q$ .

- For **Tool-Narayanaswamy-Moynihan** enter values or expressions for these properties:
  - **Reference temperature**  $T_0$ . The default is 293.15 K.
  - **Activation energy**  $Q$ .
  - **Activation energy fraction**  $\chi$  ( $0 < \chi < 1$ ).
  - **Structural relaxation times**  $\lambda_0$ .
  - **Fictive temperature weights**  $w$ .
- For **User defined** enter a value or expression for the shift function  $a_T$ .

### VISCOELASTICITY MODEL

Select a **Material model** — **Generalized Maxwell**, **Generalized Kelvin-Voigt**, **Maxwell**, **Kelvin-Voigt**, **Standard linear solid**, or **Burgers**. Then see the settings for each option that follows.

For some material models, you can select the shear modulus 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, hence they do not contribute to the stress. 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.

For linear viscoelasticity, in each **Branch** row enter the stiffness of the spring  $G_m$  in the **Shear modulus (Pa)** column and the relaxation time constant  $\tau_m$  in the **Relaxation time (s)** column for the spring-dashpot pair in branch  $m$ .

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_m$  in the **Fractional order (1)** column for each spring-spring-pot branch.

For large strain viscoelasticity, in each **Branch** row enter  $\beta_m$  (the energy factor of the branch) in the **Energy factor (1)** column and the relaxation time constant  $\tau_m$  in the **Relaxation time (s)** column for the spring-dashpot pair.

- Use the **Add** button (  ) to add a row to the table and the **Delete** button (  ) to 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 shear modulus or energy factor, and the relaxation time for that branch.

Select the **Stiffness used in stationary studies**, 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 [Linear 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.

#### *Maxwell*

For **Maxwell** enter the values for the parameter that describes the viscous behavior of the single dashpot connected in series with the spring.

Select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time**  $\tau_v$ . The default is 3000 s.
- **Viscosity**  $\eta_v$  of the dashpot. The default is  $6 \cdot 10^{13}$  Pa-s.

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_v$  of the spring-pot. The default is 0.5 (dimensionless).





Note that the instantaneous stiffness is given in the parent [Linear Elastic Material](#).

#### *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.

For linear viscoelasticity, in each **Element** row enter the stiffness of the spring  $G_m$  in the column labeled **Shear modulus (Pa)** and the relaxation time  $\tau_m$  in the column labeled **Relaxation time (s)** for the spring-dashpot pair in the element  $m$ .

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_m$  in the **Fractional order (1)** column for each spring-spring-pot branch.

- Use the **Add** button (  ) to add a row to the table and the **Delete** button (  ) to 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 elements in a text file with space-separated columns (from left to right): the element number, the shear modulus, and the relaxation time for that branch.

Select the **Stiffness used in stationary studies**, either **Long-term** or **Instantaneous**. With **Long-term** all dampers are assumed to be relaxed, hence the dampers do not contribute to any stress. 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 given in the parent [Linear Elastic 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.

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time**  $\tau_v$ . The default is 3000 s.
- **Viscosity**  $\eta_v$  of the dashpot. The default is  $6 \cdot 10^{13}$  Pa·s.

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_v$  of the spring-pot. The default is 0.5 (dimensionless).

For large strain viscoelasticity, enter the **Relaxation time**  $\tau_v$ . The default is 3000 s.

Note that the instantaneous stiffness is given in the parent [Linear 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.

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time**  $\tau_v$ . The default is 3000 s.
- **Viscosity**  $\eta_v$  of the dashpot. The default is  $6 \cdot 10^{13}$  Pa·s.

In the **Shear modulus** field, enter the stiffness of the spring  $G_v$ . The default is  $2 \cdot 10^{10}$  Pa.

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_v$  of the spring-pot. The default is 0.5 (dimensionless).

For large strain viscoelasticity, enter the **Relaxation time**  $\tau_v$ , which default value is 3000 s, and the **Energy factor**  $\beta_v$  of the dashpot. The default is 0.2.

Note that the long-term stiffness is given in the parent [Linear Elastic Material](#) or [Hyperelastic Material](#).

#### *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.

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:


- **Relaxation time.** The default is 3000 s for both dashpots  $\tau_{v1}$  and  $\tau_{v2}$ .
- **Viscosity.** Enter the viscosity of the dashpots. The default is  $6 \cdot 10^{13}$  Pa·s for both  $\eta_{v1}$  and  $\eta_{v2}$ .

In the **Shear modulus** field, enter the stiffness of the second spring  $G_{v2}$ . The default is  $2 \cdot 10^{10}$  Pa.

When the **Use fractional derivatives** check box is selected, enter the fractional order  $\beta_{v1}$  and  $\beta_{v2}$  of the spring-pot pairs. The default is 0.5 (dimensionless) for both spring-pots.

Note that the instantaneous stiffness is given in the parent [Linear Elastic Material](#).

### DISCRETIZATION

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

The check box **Use local time integration** is selected by default. Deselect it in case you want to use the global time integration scheme. The check box is only available for the Generalized Maxwell and Standard Linear Solid models. For all other viscoelasticity models, the global time integration is used.

Uncheck the **Use local time integration** check box 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 selected as one order less than what is used for the displacements. This results in that fewer extra degrees of freedom are 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** check box 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**
- *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**  
**Shell>Layered Linear Elastic Material>Viscoelasticity**  
**Layered Shell>Linear Elastic Material>Viscoelasticity**  
**Layered Shell>Hyperelastic Material>Viscoelasticity**  
**Membrane>Linear Elastic Material>Viscoelasticity**  
**Membrane>Layered Linear Elastic Material>Viscoelasticity**  
**Membrane>Nonlinear Elastic Material>Viscoelasticity**  
**Membrane>Hyperelastic Material>Viscoelasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **Nonlinear 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



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**  $d_{\infty}$ . The default is 1.
- **Damage saturation**  $W_{\text{sat}}$ . The default is 1 MJ/m<sup>3</sup>.
- **Deformation dependence coefficient**  $\alpha$ . The default is 0.

For **Miehe** enter the following settings:

- **Maximum damage**  $d_{\infty}$ . The default is 1.
- **Damage saturation**  $W_{\text{sat}}$ . The default is 1 MJ/m<sup>3</sup>.

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Hyperelastic Material>Mullins Effect**  
**Shell>Layered Hyperelastic Material>Mullins Effect**  
**Layered Shell>Hyperelastic Material>Mullins Effect**  
**Membrane>Hyperelastic Material>Mullins Effect**

### *Ribbon*

Physics tab with **Hyperelastic Material** or **Layered 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 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 Material Module or the Geomechanics Module are required for this material model, and the available options depend on the products used. For details, see <https://www.comsol.com/products/specifications/>.



See also [Elastoplastic Material Models](#) in the *Structural Mechanics Theory* chapter.

## SHELL PROPERTIES



This section is only 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.
- **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Plasticity](#) node in the [Shell and Plate](#) chapter.
- **Layered Linear Elastic Material** 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.

### *Plasticity Model*

Select **Small plastic strains** or **Large plastic strains** to apply either an additive or multiplicative decomposition between elastic and plastic strains.

- When using plasticity together with a hyperelastic material, only the option **Large plastic strains** is available.
- When using plasticity in the Shell, Membrane and Truss interfaces, only the option **Small plastic strains** is available.

### *Yield Function F*

The **Yield function F** defines the limit of the elastic regime  $F(\sigma, \sigma_{ys}) \leq 0$ .

Select a **Yield function F** criterion — **von Mises stress**, **Tresca stress**, **Hill orthotropic plasticity**, or **User defined**.

- The default is **von Mises stress** with associate plastic potential.
- Select **Tresca stress** to use a 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 plasticity** to use Hill’s criterion. For **Hill orthotropic plasticity** from the **Specify** list select either the **Initial tensile and shear yield stresses**  $\sigma_{ys0ij}$  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 **Linear Elastic Material** node.
- For **User defined** enter a different value or expression. Write any expression in terms of the stress tensor variables or its invariants in the  $\phi(\sigma)$  field.
  - For **User defined** also select the **Plastic potential**  $Q$  related to the flow rule — **Associated, von Mises,** or **User defined** (nonassociated). Enter a **User defined** value in the  $Q$  field as needed.

#### *Initial Yield Stress*

For all yield criteria, the default **Initial yield stress**  $\sigma_{ys0}$  uses values **From material** and represents the stress level where plastic deformation starts.

#### *Isotropic Hardening Model*

For all yield criteria, select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list.

- 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**  $E_{Tiso}$  uses values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is modified as hardening occurs, and it is related to the equivalent plastic strain  $\epsilon_{pe}$  as

$$\sigma_{ys} = \sigma_{ys0} + E_{iso}\epsilon_{pe} \quad \text{with} \quad \frac{1}{E_{iso}} = \frac{1}{E_{Tiso}} - \frac{1}{E}$$

For the linear isotropic hardening model, the yield stress increases proportionally to the equivalent plastic strain  $\epsilon_{pe}$ . 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  $\sigma_{ys}$  is modified by the power-law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + k\epsilon_{pe}^n$$

The **Strength coefficient**  $k$  and the **Hardening exponent**  $n$  use values **From material** (if it exists) or **User defined**.

- Select **Johnson-Cook** from the list to model strain rate dependent hardening. The **Strength coefficient**  $k$ , **Hardening exponent**  $n$ , **Reference strain rate**  $\dot{\epsilon}_0$ , 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**  $T_{\text{ref}}$ , the **Melting temperature**  $T_m$ , and the **Temperature exponent**,  $m$ .
- For **User defined**, enter the **Thermal Softening function**  $f(T_h)$ , the **Reference temperature**  $T_{\text{ref}}$ , 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

$$\sigma_{ys}(\epsilon_{pe}) = (\sigma_{ys0} + k \epsilon_{pe}^n) \left( 1 + C \log \left( \frac{\dot{\epsilon}_{pe}}{\dot{\epsilon}_0} \right) \right) (1 - f(T_h))$$

$$T_h = \frac{T - T_{\text{ref}}}{T_m - T_{\text{ref}}}$$

In the case of power law softening,  $f(T_h) = T_h^m$ .

- For **Swift** nonlinear isotropic hardening, the **Reference strain**  $\epsilon_0$  and the **Hardening exponent**  $n$  use values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is modified by the power-law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} \left( 1 + \frac{\epsilon_{pe}}{\epsilon_0} \right)^n$$

Typically, the reference strain equals the onset of plasticity,  $\epsilon_0 = \sigma_{ys0}/E$ .

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level  $\sigma_{ys}$  is modified by the exponential law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + \sigma_{\text{sat}} \left( 1 - e^{-\beta \epsilon_{pe}} \right)$$

The **Saturation flow stress**  $\sigma_{\text{sat}}$  and the **Saturation exponent**  $\beta$  use values **From material** (if it exists) or **User defined**.

- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress**  $\sigma_\infty$ , the **Saturation coefficient**  $m$ , and the **Saturation exponent**  $n$  use values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is increased by the exponential law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + (\sigma_\infty - \sigma_{ys0}) \left(1 - e^{-m\epsilon_{pe}^n}\right)$$

- For **Hardening function**, the isotropic **Hardening function**  $\sigma_h(\epsilon_{pe})$  uses values **From material** or **User defined**. The yield level  $\sigma_{ys}$  is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\epsilon_{pe})$$

- This definition implies that the hardening function  $\sigma_h(\epsilon_{pe})$  in the Material node must be zero at zero plastic strain. In other words,  $\sigma_{ys} = \sigma_{ys0}$  when  $\epsilon_{pe} = 0$ . 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  $\epsilon_{pe}$ . The variable is named using the scheme <physics>.epe, for example, solid.epe.

#### *Kinematic Hardening Model*

For all yield function criteria, select the type of kinematic hardening model (not available for hyperelastic materials) from the **Kinematic hardening model** list.

- 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  $\sigma_b$  as plasticity occurs:

$$\sigma_b = C_k \cdot \frac{2}{3} \epsilon_p \quad \text{with} \quad \frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

This is Prager's linear kinematic hardening model, so the back stress  $\sigma_b$  is collinear to the plastic strain tensor  $\epsilon_p$ .

- If **Armstrong-Frederick** is selected from the list, the default **Kinematic hardening modulus**  $C_k$  and **Kinematic hardening parameter**  $\gamma_k$  use values **From material**. These parameters are used to calculate the back stress  $\sigma_b$  from the rate equation

$$\dot{\sigma}_b = \frac{2}{3}C_k \dot{\epsilon}_p - \gamma_k \dot{\epsilon}_{pe} \sigma_b$$

This is Armstrong–Frederik nonlinear kinematic hardening model. Replacing the variable for the back strain instead of the back stress

$$\sigma_b = C_k \cdot \frac{2}{3} \epsilon_b$$

gives the rate for the back strain

$$\dot{\epsilon}_b = \dot{\epsilon}_p - \gamma_k \dot{\epsilon}_{pe} \epsilon_b$$

- 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:

$$\dot{\sigma}_{b,i} = \frac{2}{3}C_i \dot{\epsilon}_p - \gamma_i \dot{\epsilon}_{pe} \sigma_{b,i}$$





or equivalently, solve  $N$  rate equations for the back strain such as  $\sigma_{b,i} = C_i \cdot \frac{2}{3} \epsilon_{b,i}$

$$\dot{\epsilon}_{b,i} = \dot{\epsilon}_p - \gamma_i \dot{\epsilon}_{pe} \epsilon_{b,i}$$

The total back stress  $\sigma_b$  is then computed from the sum of the branches

$$\sigma_b = \frac{2}{3}C_0 \epsilon_p + \frac{2}{3} \sum_{i=1}^N C_i \epsilon_{b,i}$$

For each **Branch** row, enter  $C_i$  (the hardening modulus of the branch  $i$ ) in the **Hardening modulus (Pa)** column and  $\gamma_i$  (the hardening parameter of the branch  $i$ ) in the **Hardening parameter (I)** 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** check box 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 of **Large plastic strains**, 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**

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Plasticity**

**Solid Mechanics>Nonlinear Elastic Material>Plasticity**

**Solid Mechanics>Hyperelastic Material>Plasticity**

**Shell>Layered Linear Elastic Material>Plasticity**

**Layered Shell>Linear Elastic Material>Plasticity**

**Layered Shell>Hyperelastic Material>Plasticity**

**Membrane>Linear Elastic Material>Plasticity**

**Membrane>Layered Linear Elastic Material>Plasticity**

**Membrane>Nonlinear Elastic Material>Plasticity**

**Truss>Linear Elastic Material>Plasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

**Attributes>Plasticity**

### *Set Variables*

Use the **Set Variables** subnode to **Plasticity**, **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** 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.

### *Plasticity*

- When **Small plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic strain tensor**. 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**.
- When **Large plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic deformation gradient inverse**. From the **Plastic deformation gradient inverse** list, select **Do not set** or **User defined**. The default **User defined** value is the identity tensor.
- If **Armstrong-Frederick** or **Chaboche** is selected from the **Kinematic Hardening Model** 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**.

### *Porous Plasticity*

- When **Small plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic strain tensor**. 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**.
- When **Large plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic deformation gradient inverse**. From the **Plastic deformation gradient inverse** list, select **Do not set** or **User defined**. The default **User defined** value is the identity tensor.
- From the **Equivalent plastic strain in the matrix material** list, select **Do not set** or **User defined**. The default **User defined** value is zero.
- From the **Porosity rate** list, select **Do not set** or **User defined**. The default **User defined** value is zero.

### *Soil Plasticity, Concrete and Rocks*

- 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**.



### *Elastoplastic Soil Material*

- 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**.
- From the **Consolidation pressure rate** list, select **Do not set** or **User defined**. The default **User defined** value is zero.
- When **Extended Barcelona Basic** is selected from the **Material model** list, specify the value for the **Suction rate**. From the **Suction rate** list, select **Do not set** or **User defined**. The default **User defined** value is zero.
- When **Hardening Soil** is selected from the **Material model** list, specify the value for the **Plastic shear strain**. From the **Plastic shear strain** 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>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>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>Elastoplastic Soil Material>Set Variables**

**Shell>Layered Linear Elastic Material>Plasticity>Set Variables**

**Layered Shell>Linear Elastic Material>Plasticity>Set Variables**

**Layered Shell>Hyperplastic Material>Plasticity>Set Variables**

**Membrane>Linear Elastic Material>Plasticity>Set Variables**

**Membrane>Layered Linear Elastic Material>Plasticity>Set Variables**

**Membrane>Nonlinear Elastic Material>Plasticity>Set Variables**

**Truss>Linear Elastic Material>Plasticity>Set Variables**

### *Ribbon*

Physics tab with **Plasticity**, **Porous Plasticity**, **Soil Plasticity**, **Concrete**, **Rocks**, or **Elastoplastic Soil Material** node selected in the model tree:

## Attributes>Set Variables

### *Creep*

---

Use the **Creep** subnode to define the creep properties of the 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**, **Layered Linear Elastic Material**, and **Nonlinear Elastic Material**.

The Nonlinear Structural Material Module or the Geomechanics Module are required for this material model, and the available options depend on the products used. For details, see <https://www.comsol.com/products/specifications/>.



See also [Creep and Viscoplasticity](#) in the *Structural Mechanics Theory* chapter.

---

## SHELL PROPERTIES

---



This section is only 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.
  - **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Creep](#) node in the [Shell and Plate](#) chapter.
  - **Layered Linear Elastic Material** in the Membrane interface. See the documentation for the [Creep](#) node in the [Membrane](#) chapter.
- 

## CREEP DATA

### *Nonlinear Structural Materials Module*

Select a **Material model** — **Norton**, **Norton-Bailey**, **Garofalo (hyperbolic sine)**, **Nabarro-Herring**, **Coble**, **Weertman**, **Potential**, **Volumetric**, **Deviatoric**, or **User defined**. Then follow the instructions as below.

### *Geomechanics Module*

Select a **Material model** — **Potential**, **Volumetric**, **Deviatoric**, or **User defined**. Then follow the instructions as below.

### *Norton*

For **Norton** enter the following settings:

- **Creep rate coefficient**  $A$ .
- **Reference creep stress**  $\sigma_{\text{ref}}$ . The default is 1 MPa.
- **Stress exponent**  $n$ .
- Select the **Include temperature dependency** check box to add an “Arrhenius-type” temperature dependence. Then enter a **Creep activation energy**  $Q$ .

### *Norton–Bailey*

For **Norton–Bailey** enter the following settings:

- **Creep rate coefficient**  $A$ .
- **Reference creep stress**  $\sigma_{\text{ref}}$ . The default is 1 MPa.
- **Stress exponent**  $n$ .
- Select a **Hardening model** — **Time hardening** or **Strain hardening**.
  - For **Time hardening**, enter the **hardening exponent**  $m$ , the **Time shift**  $t_{\text{shift}}$  and the **Reference time**  $t_{\text{ref}}$ .
  - For **Strain hardening**, enter the **hardening exponent**  $m$ , the **Equivalent creep strain shift**  $\varepsilon_{\text{shift}}$ , and the **Reference time**  $t_{\text{ref}}$ .
- Select the **Include temperature dependency** check box to add an “Arrhenius-type” temperature dependence. Then enter a **Creep activation energy**  $Q$ .

### *Garofalo (Hyperbolic Sine)*

For **Garofalo (hyperbolic sine)** enter the following settings:

- **Creep rate coefficient**  $A$ .
- **Reference creep stress**  $\sigma_{\text{ref}}$ . The default is 1 MPa.
- **Garofalo n parameter**  $n$ .
- Select the **Include temperature dependency** check box as needed. Then enter a **Creep activation energy**  $Q$ .

### *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:

- **Ionic diffusivity**  $D_{gb}$ .
- **Burgers vector**  $b$ .
- **Grain diameter**  $d$ .

### *Weertman*

For **Weertman** enter the following settings:

- **Diffusivity**  $D_0$ .
- **Burgers vector**  $b$ .
- **Stress exponent**  $n$ .
- **Reference creep stress**  $\sigma_{ref}$ . The default is 1 MPa.

### *Potential*

For **Potential** enter the following settings:

- **Rate multiplier**  $\eta$ .
- **Creep potential**  $Q_{cr}$ .

### *Deviatoric*

For **Deviatoric** enter the **Equivalent creep strain rate**  $\epsilon_{cre}$ .

### *Volumetric*

For **Volumetric** enter the **Volumetric creep strain rate**  $\epsilon_{cr,vol}$ .

### *User Defined*

For **User defined** enter each element for the symmetric **Creep strain rate tensor**  $\epsilon_{cr}$ . The tensor components are interpreted in the coordinate system of the parent node.



To compute the energy dissipation caused by creep, enable the **Calculate dissipated energy** check box in the [Energy Dissipation](#) section of the parent material node (**Linear Elastic Material** or **Nonlinear Elastic Material**).

---



For an example of **Norton** and **Norton-Baily** material models, see *Combining Creep Material Models*: Application Library path **Nonlinear\_Structural\_Materials\_Module/Creep/combined\_creep**.

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Creep**

**Solid Mechanics>Nonlinear Elastic Material>Creep**

**Shell>Layered Linear Elastic Material>Creep**

**Layered Shell>Linear Elastic Material>Creep**

**Membrane>Linear Elastic Material>Creep**

**Membrane>Layered Linear Elastic Material>Creep**

**Membrane>Nonlinear Elastic Material>Creep**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

**Attributes>Creep**

### *Viscoplasticity*

---

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model. This material model is available in the Solid Mechanics, Layered Shell, Shell, and Membrane interfaces, and can be used together with **Linear Elastic Material**, **Layered Linear Elastic Material**, and **Nonlinear Elastic Material**.

The Nonlinear Structural Material Module is required for this material model.



See also [Creep and Viscoplasticity](#) in the *Structural Mechanics Theory* chapter.

---

## SHELL PROPERTIES



This section is only 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.
- **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Viscoplasticity](#) node in the [Shell and Plate](#) chapter.
- **Layered Linear Elastic Material** in the Membrane interface. See the documentation for the [Viscoplasticity](#) node in the [Membrane](#) chapter.

## VISCOPLASTICITY MODEL

Select a **Viscoplasticity model** — **Anand**, **Chaboche**, or **Perzyna**. Then follow the instructions as below.

### *Anand*

For **Anand** enter the following data:

- **Creep rate coefficient**  $A$ .
- **Creep activation energy**  $Q$ .
- **Multiplier of stress**  $\xi$ .
- **Stress sensitivity**  $m$ .
- **Deformation resistance saturation coefficient**  $s_0$ .
- **Deformation resistance initial value**  $s_{\text{init}}$ .
- **Hardening constant**  $h_0$ .
- **Hardening sensitivity**  $\alpha$ .
- **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**

### Chaboche

For **Chaboche** enter the following settings:

- **Viscoplastic rate coefficient**  $A$ .
- **Reference creep stress**  $\sigma_{ref}$ . The default is 1 MPa.
- **Stress exponent**  $n$ .

### Yield Function $F$

The **Yield function  $F$**  defines the limit of the elastic regime  $F(\sigma, \sigma_{ys}) \leq 0$ .

Select a **Yield function  $F$**  criterion — **von Mises stress**, **Tresca stress**, **Hill orthotropic plasticity**, or **User defined**.

- The default is **von Mises stress** with associate plastic potential.
- Select **Tresca stress** to use a 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 plasticity** to use Hill's criterion. For **Hill orthotropic plasticity** from the **Specify** list select either the **Initial tensile and shear yield stresses**  $\sigma_{ys0ij}$  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 feature.
- For **User defined** enter a different value or expression. In the  $\phi(\sigma)$  field write any expression in terms of the stress tensor components or its invariants.
  - For **User defined** also select the **Plastic potential  $Q$**  related to the flow rule — **Associated**, **von Mises**, or **User defined** (nonassociated). Enter a **User defined** value in the  $Q$  field as needed.

### Initial Yield Stress

The default **Initial yield stress**  $\sigma_{ys0}$  uses values **From material** and represents the stress level where viscoplastic deformation starts.

### Isotropic Hardening Model

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list.

- 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**  $E_{Tiso}$  uses values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is modified as hardening occurs, and it is related to the equivalent viscoplastic strain  $\epsilon_{vpe}$  as

$$\sigma_{ys} = \sigma_{ys0} + E_{iso}\epsilon_{vpe} \quad \text{with} \quad \frac{1}{E_{iso}} = \frac{1}{E_{T_{iso}}} - \frac{1}{E}$$

For the linear isotropic hardening model, the yield stress increases proportionally to the equivalent viscoplastic strain  $\epsilon_{vpe}$ . 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  $\sigma_{ys}$  is modified by the power-law

$$\sigma_{ys} = \sigma_{ys0} + k\epsilon_{vpe}^n$$

the **Strength coefficient**  $k$  and the **Hardening exponent**  $n$  use values **From material** (if it exists) or **User defined**.

- Select **Johnson-Cook** from the list to model strain rate dependent hardening. The **Strength coefficient**  $k$ , **Hardening exponent**  $n$ , **Reference strain rate**  $\dot{\epsilon}_0$ , 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 has the properties  $f(0) = 0$  and  $f(1) = 1$ . The variable is named using the scheme `<physics>.<elasticTag>.<viscoplasticTag>.Th` (for example `solid.lemm1.vp11.Th`).

The yield stress and hardening function for the Johnson–Cook model is given by

$$\sigma_{ys}(\epsilon_{pe}) = (\sigma_{ys0} + k\epsilon_{vpe}^n) \left( 1 + C \log \left( \frac{\dot{\epsilon}_{vpe}}{\dot{\epsilon}_0} \right) \right) (1 - f(T_h))$$

$$T_h = \frac{T - T_{ref}}{T_m - T_{ref}}$$

In the case of power law softening,  $f(T_h) = T_h^m$ .

- For **Swift** nonlinear isotropic hardening, the **Reference strain**  $\epsilon_0$  and the **Hardening exponent**  $n$  use values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is modified by the power-law



$$\sigma_{ys} = \sigma_{ys0} \left( 1 + \frac{\epsilon_{vpe}}{\epsilon_0} \right)^n$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level  $\sigma_{ys}$  is modified by the exponential law

$$\sigma_{ys} = \sigma_{ys0} + \sigma_{sat} \left( 1 - e^{-\beta \epsilon_{vpe}} \right)$$

the **Saturation flow stress**  $\sigma_{sat}$  and the **Saturation exponent**  $\beta$  use values **From material** (if it exists) or **User defined**.

- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress**  $\sigma_{\infty}$ , the **Saturation coefficient**  $m$ , and the **Saturation exponent**  $n$  use values **From material** (if it exists) or **User defined**. The yield level  $\sigma_{ys}$  is increased by the exponential law

$$\sigma_{ys} = \sigma_{ys0} + (\sigma_{\infty} - \sigma_{ys0}) \left( 1 - e^{-m \epsilon_{vpe}^n} \right)$$

- For **Hardening function**, the isotropic **Hardening function**  $\sigma_h(\epsilon_{vpe})$  uses values **From material** or **User defined**. The yield level  $\sigma_{ys}$  is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\epsilon_{vpe})$$

This definition implies that the hardening function  $\sigma_h(\epsilon_{vpe})$  in the Material node must be zero at zero viscoplastic strain. In other words,  $\sigma_{ys} = \sigma_{ys0}$  when  $\epsilon_{vpe} = 0$ . 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  $\epsilon_{vpe}$ . The variable is named using the scheme `<physics>.evpe`, for example, `solid.evpe`.

#### *Kinematic Hardening Model*

Select the type of kinematic hardening model from the **Kinematic hardening model** list.

- 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  $\sigma_b$  as:

$$\sigma_b = C_k \cdot \frac{2}{3} \varepsilon_{vp} \quad \text{with} \quad \frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

This is Prager's linear kinematic hardening model, so the back stress  $\sigma_b$  is collinear to the viscoplastic strain tensor  $\varepsilon_{vp}$ .

- If **Armstrong-Frederick** is selected from the list, the default **Kinematic hardening modulus**  $C_k$  and **Kinematic hardening parameter**  $\gamma_k$  use values **From material**. These parameters are used to calculate the back stress  $\sigma_b$  from the rate equation





$$\dot{\sigma}_b = \frac{2}{3} C_k \dot{\varepsilon}_{vp} - \gamma_k \dot{\varepsilon}_{vpe} \sigma_b$$

This is Armstrong–Frederik 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  $\gamma_i$  (the hardening parameter of the branch  $i$ ) in the **Hardening parameter (I)** 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  $\sigma_b$  is then computed from the sum

$$\sigma_b = \frac{2}{3} C_0 \varepsilon_{vp} + \sum_{i=1}^N \sigma_{b,i}$$

*Perzyna*

For **Perzyna** enter the following settings:

- **Viscoplastic rate coefficient**  $A$ .
- **Reference creep stress**  $\sigma_{ref}$ . The default is 1 MPa.

The other settings are the same as for *Chaboche*.



To compute the energy dissipation caused by viscoplasticity, enable the **Calculate dissipated energy** check box in the [Energy Dissipation](#) section of the parent material node (**Linear Elastic Material** or **Nonlinear Elastic Material**).

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Viscoplasticity**  
**Solid Mechanics>Nonlinear Elastic Material>Viscoplasticity**  
**Shell>Layered Linear Elastic Material>Viscoplasticity**  
**Layered Shell>Linear Elastic Material>Viscoplasticity**  
**Membrane>Linear Elastic Material>Viscoplasticity**  
**Membrane>Layered Linear Elastic Material>Viscoplasticity**  
**Membrane>Nonlinear Elastic Material>Viscoplasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

**Attributes>Viscoplasticity**

### *Porous Plasticity*

Use the **Porous Plasticity** subnode to define the properties of a plasticity model for a porous material.

## POROUS PLASTICITY MODEL

Use this section to define the plastic properties of the porous material.

### *Plasticity Model*

Select **Small plastic strains** or **Large plastic strains** to apply either an additive or multiplicative decomposition between elastic and plastic strains.

### *Yield Function $F$*

The **Yield function  $F$**  defines the limit of the elastic regime  $F(\sigma, \sigma_{ys}) \leq 0$ .

Select a **Yield function F** for the porous plasticity criterion — **Shima-Oyane**, **Gurson**, **Gurson-Tvergaard-Needleman**, **Fleck-Kuhn-McMeeking**, **FKM-GTN**, or **Capped Drucker-Prager**.

*Shima–Oyane*

For **Shima-Oyane** enter the following data:

- **Initial yield stress**  $\sigma_{ys0}$ .
- **Shima-Oyane parameter**  $\alpha$ .
- **Shima-Oyane parameter**  $\gamma$ .
- **Shima-Oyane parameter**  $m$ .
- **Initial void volume fraction**  $f_0$ .

*Gurson*

For **Gurson** enter the following data:

- **Initial yield stress**  $\sigma_{ys0}$ .
- **Initial void volume fraction**  $f_0$ .

*Gurson–Tvergaard–Needleman*

For **Gurson-Tvergaard-Needleman** enter the following data:

- **Initial yield stress**  $\sigma_{ys0}$ .
- **Tvergaard correction coefficient**  $q_1$ .
- **Tvergaard correction coefficient**  $q_2$ .
- **Initial void volume fraction**  $f_0$ .
- **Critical void volume fraction**  $f_c$ .
- **Failure void volume fraction**  $f_f$ .

*Fleck–Kuhn–McMeeking*

For **Fleck-Kuhn-McMeeking** enter the following data:

- **Initial yield stress**  $\sigma_{ys0}$ .
- **Initial void volume fraction**  $f_0$ .
- **Maximum void volume fraction**  $f_{max}$ .

*FKM–GTN*

For **FKM-GTN** enter the following data:

- **Initial yield stress**  $\sigma_{ys0}$ .

- **Tvergaard correction coefficient**  $q_1$ .
- **Tvergaard correction coefficient**  $q_2$ .
- **Initial void volume fraction**  $f_0$ .
- **Maximum void volume fraction**  $f_{\max}$ .
- **Void volume fraction cutoff for GTN model**  $f_{\text{GTN}}$ .
- **Void volume fraction cutoff for FKM model**  $f_{\text{FKM}}$ .

#### *Capped Drucker–Prager*

For **Capped Drucker-Prager** enter the following data:

- **Drucker-Prager alpha coefficient**  $\alpha$ .
- **Drucker-Prager k coefficient**  $k$ .
- **Initial void volume fraction**  $f_0$ .

The material properties use values **From material** (default) or **User defined**.

#### *Void Growth*

It is possible to **Include void nucleation in tension** or **Include void growth in shear** by selecting the corresponding check box. See the section [Void Growth](#) for details.

#### *Isotropic Hardening Model*

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list.

- Select **Perfectly plastic** (ideal plasticity) if the material can undergo plastic deformation without any increase in yield stress. When **Capped Drucker-Prager** is selected, enter values or expressions to define the semi-axes of the cap under **Elliptic cap parameter**  $p_a$  and **Elliptic cap parameter**  $p_b$ .
- For **Linear** the default **Isotropic tangent modulus**  $E_{T\text{iso}}$  uses values **From material** (if it exists) or **User defined**. The flow stress (yield level)  $\sigma_{\text{fm}}$  is modified as hardening occurs, and it is related to the equivalent plastic strain in the porous matrix  $\epsilon_{\text{pm}}$  as

$$\sigma_{\text{fm}} = \sigma_{\text{ys0}} + E_{\text{iso}}\epsilon_{\text{pm}} \quad \text{with} \quad \frac{1}{E_{\text{iso}}} = \frac{1}{E_{T\text{iso}}} - \frac{1}{E}$$

For the linear isotropic hardening model, the flow stress (yield stress) increases proportionally to the equivalent plastic strain in the porous matrix  $\epsilon_{\text{pm}}$ . 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)  $\sigma_{\text{fm}}$  is modified by the power-law

$$\sigma_{fm} = \sigma_{ys0} + k\epsilon_{pm}^n$$

The **Strength coefficient**  $k$  and the **Hardening exponent**  $n$  use values **From material** (if it exists) or **User defined**.

- For **Power law** isotropic hardening, the **Hardening exponent**  $n$  uses the value **From material** (if it exists) or **User defined**. The flow stress (yield level)  $\sigma_{fm}$  is modified by the power-law

$$\epsilon = \frac{\sigma_{ys0}}{E} \left( \frac{\sigma_{fm}}{\sigma_{ys0}} \right)^n \quad \text{for } \sigma_{fm} > \sigma_{ys0}$$

The Young's modulus  $E$  is taken from the elastic material properties.

- For **Hardening function**, the isotropic **Hardening function**  $\sigma_h(\epsilon_{pm})$  uses values **From material** or **User defined**. The flow stress (yield level)  $\sigma_{fm}$  is modified as

$$\sigma_{fm} = \sigma_{ys0} + \sigma_h(\epsilon_{pm})$$

- This definition implies that the hardening function  $\sigma_h(\epsilon_{pm})$  in the Material node must be zero at zero plastic strain. In other words,  $\sigma_{fm} = \sigma_{ys0}$  when  $\epsilon_{pm} = 0$ . 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  $\epsilon_{pm}$ . The variable is named using the scheme `<physics>.<elasticTag>.<plasticTag>.epm`, for example, `solid.lemm1.popl1.epm`.
- For **Exponential** hardening, the cap in the **Capped Drucker-Prager** model evolves with the volumetric strain. Since the volumetric plastic strain  $\epsilon_{pvol}$  is negative in compression, the limit pressure  $p_b$  in the cap increases from  $p_{b0}$  as hardening evolves

$$p_b = p_{b0} + K_{iso} \log \left( 1 + \frac{\epsilon_{pvol}}{\epsilon_{pvol,max}} \right)$$

The **Isotropic hardening modulus**  $K_{iso}$ , the **Maximum plastic volumetric strain**  $\epsilon_{pvol,max}$  and the **Ellipse aspect ratio**  $R$  use values **From material** (if it exists) or **User defined**. Enter a value or expression to define the initial semi-axis of the ellipse under the **Initial location of the cap**  $p_{b0}$ .



See also [Porous Plasticity](#), [Elliptic Cap](#), and [Elliptic Cap With Hardening](#) in the *Structural Mechanics Theory* chapter.

---



---

To compute the energy dissipation caused by porous compaction, enable the **Calculate dissipated energy** check box in the [Energy Dissipation](#) section of the parent material node (**Linear Elastic Material** or **Nonlinear Elastic Material**).

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Porous Plasticity**

**Solid Mechanics>Nonlinear Elastic Material>Porous Plasticity**

**Membrane>Linear Elastic Material>Porous Plasticity**

**Membrane>Nonlinear Elastic Material>Porous Plasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic 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 plasticity. This material model can be used together with [Linear Elastic Material](#) and [Nonlinear Elastic Material](#). It is available with the Geomechanics Module.

The yield criteria are described in the theory section:

- [The Drucker–Prager Criterion](#)
- [The Mohr–Coulomb Criterion](#)
- [The Matsuoka–Nakai Criterion](#)
- [The Lade–Duncan Criterion](#)

## SOIL PLASTICITY

Select a **Yield criterion** — **Drucker-Prager**, **Mohr-Coulomb**, **Matsuoka-Nakai**, or **Lade-Duncan**. Most values are taken **From material**. For **User defined** choices, enter other values or expressions.

### *Drucker–Prager*

In the standard Drucker–Prager formulation, the material parameters are given in terms of the  $\alpha$  and  $k$  coefficients. Often material data is expressed in the parameters  $c$

and  $\phi$  used in the Mohr–Coulomb model. You can then choose to use these parameters instead. If so, select the **Match to Mohr–Coulomb criterion** check box (see [The Mohr–Coulomb Criterion](#)). If this check box is selected, the default values for **Cohesion**  $c$  and the **Angle of internal friction**  $\phi$  are taken **From material**.

If required, select the **Use dilatation angle in plastic potential** check box. If this check box is selected, then enter a value or expression for the **Dilatation angle**  $\psi$ . Alternatively, select **From material**. The dilatation angle replaces the angle of internal friction when defining the plastic potential.

If the **Match to Mohr–Coulomb criterion** check box is *not* selected, then the default **Drucker-Prager alpha coefficient** and **Drucker-Prager k coefficient** are taken **From material**.

If required, select the **Include elliptic cap** check box. Select from the list the hardening model. When **Perfectly plastic** (no hardening) is selected, enter values or expressions to define the semi-axes of the ellipse under **Elliptic cap parameter**  $p_a$  and **Elliptic cap parameter**  $p_b$ . When **Isotropic hardening** is selected from the list, the default **Isotropic hardening modulus**  $K_{iso}$ , the **Maximum plastic volumetric strain**  $\epsilon_{pvol,max}$ , and the **Ellipse aspect ratio**  $R$  are taken **From material** (see [Elliptic Cap With Hardening](#)). Enter a value or expression to define the initial semi-axis of the ellipse under the **Initial location of the cap**  $p_{b0}$ .

#### *Mohr–Coulomb*

The default **Angle of internal friction**  $\phi$  and **Cohesion**  $c$  are taken **From material**.

If required, select the **Use dilatation angle in plastic potential** check box. If this check box is selected, then enter a value or expression for the **Dilatation angle**  $\psi$ . Alternatively, select **From material**. The dilatation angle replaces the angle of internal friction when defining the plastic potential.

Under **Plastic potential** select either **Drucker-Prager matched at compressive meridian**, **Drucker-Prager matched at tensile meridian**, or **Associated**.

If required, select the **Include elliptic cap** check box. Select from the list the hardening model. When **Perfectly plastic** (no hardening) is selected, enter values or expressions to define the semi-axes of the ellipse under **Elliptic cap parameter**  $p_a$  and **Elliptic cap parameter**  $p_b$ . When **Isotropic hardening** is selected from the list, the default **Isotropic hardening modulus**  $K_{iso}$ , the **Maximum plastic volumetric strain**  $\epsilon_{pvol,max}$ , and the **Ellipse aspect ratio**  $R$  are taken **From material** (see [Elliptic Cap With Hardening](#)). Enter a value or expression to define the initial semi-axis of the ellipse under the **Initial location of the cap**  $p_{b0}$ .



### *Matsuoka–Nakai*

If required, select the **Match to Mohr-Coulomb criterion** check box. If this check box is selected, the default **Angle of internal friction**  $\phi$  is taken **From material**.

If the **Match to Mohr-Coulomb criterion** check box is *not* selected, then the default **Matsuoka-Nakai mu coefficient**  $\mu$  is taken **From material**.

### *Lade–Duncan*

If required, select the **Match to Mohr-Coulomb criterion** check box. If this check box is selected, then enter a value or expression for the **Angle of internal friction**  $\phi$ .

Alternatively, select **From material**.

If the **Match to Mohr-Coulomb criterion** check box is *not* selected, then the default **Lade-Duncan k coefficient**  $k$  is taken **From material**.

## TENSION CUTOFF

This section is only available with the **Drucker-Prager** and **Mohr-Coulomb** models

If required, select either **None**, **Mean stress cutoff**, or **Principal stress cutoff** (see [Tension Cutoff](#)).

When **Mean stress cutoff** is selected from the list, enter a value or expression for the **Maximum mean stress**  $\sigma_m$ . Use this to constrain the soil plasticity model with an extra yield surface which limits the maximum pressure in tension.

When **Principal stress cutoff** is selected from the list, enter a value or expression for the **Maximum tensile stress**  $\sigma_t$ . Use this to constrain the soil plasticity model with an extra yield surface, which limits the maximum principal tensile stress.



- *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**
- 

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Soil Plasticity**

**Solid Mechanics>Nonlinear Elastic Material>Soil Plasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic Material** node selected in the model tree:

### **Attributes>Soil plasticity**

### *Concrete*

---

In the **Concrete** subnode you define the properties for modeling materials with failure criteria representative of concrete. This material model can be used together with [Linear Elastic Material](#) and [Nonlinear Elastic Material](#). It is available with the Geomechanics Module.

The failure criteria are described in the theory section:

- [The Bresler–Pister Criterion](#)
- [The Willam–Warnke Criterion](#)
- [The Ottosen Criterion](#)

### **CONCRETE MODEL**

Select a **Concrete criterion** — **Bresler-Pister**, **Willam-Warnke**, or **Ottosen**. The defaults are taken **From material**. For **User defined** choices, enter other values or expressions.

#### *Bresler–Pister*

The defaults for the **Uniaxial tensile strength**  $\sigma_t$ , **Uniaxial compressive strength**  $\sigma_c$ , and **Biaxial compressive strength**  $\sigma_b$  are taken **From material**.

#### *Willam–Warnke*

The defaults for the **Uniaxial tensile strength**  $\sigma_t$ , **Uniaxial compressive strength**  $\sigma_c$ , and **Biaxial compressive strength**  $\sigma_b$  are taken **From material**.

#### *Ottosen*

The defaults for the **Uniaxial tensile strength**  $\sigma_c$ , Ottosen’s parameters  $a$  and  $b$ , **Size factor**  $k_1$ , and **Shape factor**  $k_2$  are taken **From material**.

### **TENSION CUTOFF**

If required, select either **None** or **Principal stress cutoff** (see [Tension Cutoff](#)).

When **Principal stress cutoff** is selected from the list, enter a value or expression for the **Maximum tensile stress**  $\sigma_t$ . Use this to constrain the concrete model with an extra yield surface, which limits the maximum principal tensile stress.

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Linear Elastic Material>Concrete**

**Solid Mechanics>Nonlinear 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 failure criteria are described in the theory section:

- [The Original Hoek–Brown Criterion](#)
- [The Generalized Hoek–Brown Criterion](#)

## ROCK MODEL

Select a **Rock criterion** — **Original Hoek–Brown** or **Generalized Hoek–Brown**. The defaults are taken **From material**. For **User defined** choices, enter other values or expressions.

### *Original Hoek–Brown*

The defaults for the **Uniaxial compressive strength**  $\sigma_c$ , **Hoek–Brown m parameter**  $m$ , and **Hoek–Brown s parameter**  $s$  are taken **From material**.

### *Generalized Hoek–Brown*

The defaults for the **Uniaxial compressive strength**  $\sigma_c$ , **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** (see [Tension Cutoff](#)).

When **Principal stress cutoff** is selected from the list, enter a value or expression for the **Maximum tensile stress**  $\sigma_t$ . Use this to constrain the rock model with an extra yield surface, which limits the maximum principal tensile stress.

## 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**

### *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 <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES



This section is only 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.
- **Layered Linear Elastic Material** 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 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**  $T_{\text{ref}}$  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 (ht/sol1)**), if any temperature variables exist, or manually enter a value or 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 select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

where  $\alpha$  is the secant coefficient of thermal expansion.  $\alpha$  can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\epsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where  $\alpha_t$  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>Magnetostrictive 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>Layered Linear Elastic Material>Thermal Expansion**  
**Membrane>Nonlinear Elastic Material>Thermal Expansion**  
**Membrane>Hyperelastic Material>Thermal Expansion**  
**Multibody Dynamics>Linear Elastic Material>Thermal Expansion**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Hyperelastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Piezoelectric Material**, or **Magnetostrictive 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

$$\epsilon_{hs} = \beta_h M_m (c_{mo} - c_{mo,ref})$$

where  $\beta_h$  is the coefficient of hygroscopic swelling,  $M_m$  is the molar mass,  $c_{mo}$  is the moisture concentration, and  $c_{mo,ref}$  is the strain-free reference concentration.

The **Hygroscopic Swelling** subnode is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES

---



This section is only 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.
  - **Layered Linear Elastic Material** 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 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**  $c_{\text{ref}}$ . 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**  $\beta_h$  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  $\beta_h$ . The default value for the **User defined** case is  $1.5e-4 \text{ m}^3/\text{kg}$ . 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** check box 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**  
**Layered Shell>Linear Elastic Material>Hygroscopic Swelling**  
**Layered Shell>Hyperelastic Material>Hygroscopic Swelling**  
**Membrane>Linear Elastic Material>Hygroscopic Swelling**  
**Membrane>Layered Linear Elastic Material>Hygroscopic Swelling**  
**Membrane>Nonlinear Elastic Material>Hygroscopic Swelling**  
**Membrane>Hyperelastic Material>Hygroscopic Swelling**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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 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 <https://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**  $\boldsymbol{\varepsilon}_0$ . 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  $\varepsilon_{0x}$ ,  $\varepsilon_{0y}$ , and  $\varepsilon_{0z}$  and off-diagonal components  $\varepsilon_{0xy}$ ,  $\varepsilon_{0yz}$ , and  $\varepsilon_{0xz}$ , 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 Stresses and Strains](#).
- 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>Magnetostrictive Material>Initial Stress and Strain**

**Solid Mechanics>Shape Memory Alloy>Initial Stress and Strain**

**Layered Shell>Linear Elastic Material>Initial Stress and Strain**

**Multibody Dynamics>Linear Elastic Material>Initial Stress and Strain**

### Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Piezoelectric Material**, **Magnetostrictive 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 <https://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.

---

### SHELL PROPERTIES

---



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)**, **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**  $S_{\text{ext}}$  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** or **Load contribution only** — to determine the effect of the contribution.

- 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**  $\sigma_{\text{ext}}$  as **Isotropic**, **Diagonal**, or **Symmetric**. Choose a **Contribution type** — **Add to stress tensor** or **Load contribution only** — to determine the effect of the contribution.
- When **Pore pressure** is selected, the **Absolute pressure** drop down menu 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**) which compute and announce a pressure variable, such variables are also present in the list. You can also enter a **Reference pressure level**  $p_{\text{ref}}$ , which is the pressure level at which the pore pressure does not give any stress contribution. Enter the **Biot-Willis coefficient**  $\alpha_B$  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**  $\sigma_{\text{ext}}$  as **Isotropic**,

**Diagonal**, or **Symmetric**. When using **in situ stress**, there are both contributions to the stress tensor, as well as a body load.



- 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**  
**Layered Shell>Linear Elastic Material>External Stress**  
**Layered Shell>Hyperelastic 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](#), and [Hyperelastic Material](#) on a variety of formats, including using external coded functions.

## 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**, or **Stretches**.

- 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 which 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  $\epsilon_{\text{ext}}$ . From the **Strain tensor** list, you can choose **User defined**, or any strain tensor which 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  $F_{\text{ext}}$ . 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  $F_{\text{ext}}^{-1}$ . 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.



The **External Strain** node is only available with some COMSOL products (see <https://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** from the **Strain input** list, and the **Additive strain decomposition** check-box 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** from 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 **Force Linear Strains** check-box 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**

**Shell>Layered Hyperelastic Material>External Strain**

**Layered Shell>Linear Elastic Material>External Strain**

**Layered Shell>Hyperelastic 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**

### *Damage*

Use the **Damage** subnode to model damage and cracking in brittle materials according to various criteria. It is available with the **Linear Elastic Material** in the Solid Mechanics and Layered Shell interfaces, and with the **Layered Linear Elastic Material** in the Shell interface.

## SHELL PROPERTIES



This section is only 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.
- **Layered Linear Elastic Material** 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** check box 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**  $\sigma_{ts}$ , the default 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 **Damage evolution** function  $d(\kappa)$ .

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**  $\varepsilon_f$  accordingly. The available options depend on the spatial regularization method selected.

For **Multilinear strain softening**, also enter the **Shape factor**  $\lambda$ . Its default value is 0.5.

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**  $\tau$ . 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**  $\beta$ , 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**  $\sigma_{ts}$ , 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**  $\epsilon_{ft}$  accordingly.
- For **Mazars damage evolution function**, enter the **Tensile strain threshold**  $\epsilon_{0t}$ , 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**  $\epsilon_{0c}$ , 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 **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**  $\tau$ . 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** in the Solid Mechanics interface. See [Phase Field Damage Models](#) for details.

Select the type of **Crack driving force** — **Strain energy density** or **Principal stress criterion**. See [Crack Driving Force](#) for details.

- For **Strain energy density**, enter a value for the **Critical energy release rate**  $G_c$ , and the **Strain energy threshold**  $G_{c0}$ .
- For **Principal stress criterion**, enter a value for the **Critical fracture stress**  $\sigma_c$ , and the **Post-peak slope parameter**  $\xi$ .

Enter a value for the **Internal 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 **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**. See [Strain Energy Split](#) for details.

The **Viscous regularization** check box is not selected by default. When selected, a viscous term is added to the evolution of the crack phase field in time dependent studies. When selected, enter a value for the **Characteristic time**  $\tau$ . See [Viscous Regularization](#) for details.

#### **ADVANCED**

Enter the **Maximum damage**. The default value is 0.99999.

## DISCRETIZATION

This section is available with the **Phase field damage** model. Select the Lagrange order of the **Crack phase field  $\phi$**  — **Same as displacement field**, **Linear**, **Quadratic**, **Cubic**, **Quartic**, or **Quintic**.

The **Damage** subnode is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).



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**

**Shell>Layered Linear Elastic Material>Damage**

**Layered Shell>Linear Elastic Material>Damage**

### *Ribbon*

Physics tab with **Linear Elastic Material** or **Layered Linear Elastic Material**, 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**.

When the material is activated, the following is done to ensure that the activation is stress-free:

- Elastic strains at the instant of activation are removed from the total strains.
- Initial stresses are always removed, regardless of the **Activation expression**.

The **Activation** subnode can be used in combination with **Linear Elastic Material**.

### 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 <https://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**

**Layered Shell>Linear Elastic Material>Activation**

**Membrane>Linear Elastic Material>Activation**

**Truss>Linear Elastic Material>Activation**

#### *Ribbon*

Physics tab with **Linear Elastic Material**, node selected in the model tree:

**Attributes>Activation**

## 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**, **Nonlinear Elastic Material**, and **Layered Linear Elastic Material**. 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 <https://www.comsol.com/products/specifications/>)



## SHELL PROPERTIES



This section is only 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.
- **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Safety](#) node in the [Shell and Plate](#) chapter.
- **Layered Linear Elastic Material** 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
St. Venant	X	X	X	X	X	X
Mohr-Coulomb	X	–	–	–	–	–
Drucker-Prager	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	–	–
Norris	Plane stress	X	–	X	–	–
Modified Tsai-Hill	Plane stress	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 <sup>1</sup>	–	X	X	–	–	–
Hashin-Rotem <sup>1</sup>	–	X	X	–	–	–
Hashin <sup>1</sup>	–	X	X	–	–	–
Puck <sup>1</sup>	–	X	X	–	–	–
LaRC03 <sup>1</sup>	–	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**  $\sigma_{ts}$ .
- When **Failure Criterion** is **Tresca**, enter **Tensile strength**  $\sigma_{ts}$ .
- When **Failure Criterion** is **Rankine**, enter **Tensile strength**  $\sigma_{ts}$  and **Compressive strength**  $\sigma_{cs}$ .
- When **Failure Criterion** is **St. Venant**, enter **Ultimate tensile strain**  $\epsilon_{ts}$  and **Ultimate compressive strain**  $\epsilon_{cs}$ .
- 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**  $\phi$ .  
 When **Tensile and compressive strengths** is used, enter **Tensile strength**  $\sigma_{ts}$  and **Compressive strength**  $\sigma_{cs}$ .  
 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**  $\alpha$  and **Drucker-Prager k coefficient**  $k$ .  
 When **Tensile and compressive strengths** is used, enter **Tensile strength**  $\sigma_{ts}$  and **Compressive strength**  $\sigma_{cs}$ .

When **Mohr-Coulomb parameters** is used, enter **Cohesion**  $c$  and **Angle of internal friction**  $\phi$ .

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 **Bresler-Pister**, enter **Tensile strength**  $\sigma_{ts}$ , **Compressive strength**  $\sigma_{cs}$ , and **Biaxial compressive strength**  $\sigma_{bc}$ .
- When **Failure Criterion** is **Willam-Warnke**, enter **Tensile strength**  $\sigma_{ts}$ , **Compressive strength**  $\sigma_{cs}$ , and **Biaxial compressive strength**  $\sigma_{bc}$ .
- When **Failure Criterion** is **Ottosen**, enter the **Compressive strength**  $\sigma_{cs}$ , **Ottosen parameters**  $a$  and  $b$ , the **Size factor**  $k_1$ , and the **Shape factor**  $k_2$ .
- When **Failure Criterion** is **Jenkins**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Waddoups**, enter **Ultimate tensile strains**  $\varepsilon_{ts}$ , **Ultimate compressive strains**  $\varepsilon_{cs}$ , and **Ultimate shear strains**  $\gamma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Azzi-Tsai-Hill**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Norris**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Modified Tsai-Hill**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Tsai-Hill**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hoffman**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Tsai-Wu Orthotropic**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.

- When **Failure Criterion** is **Zinoviev**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hashin-Rotem**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hashin**, enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Puck**:
  - Enter **Tensile strengths**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy.
  - Enter the **Fiber failure** data: **Ultimate tensile strain in longitudinal direction**,  $\epsilon_{ts1}$ , and **Ultimate compressive strain in longitudinal direction**,  $\epsilon_{cs1}$ . Also, enter the fiber material properties **Young's modulus of fiber in longitudinal direction**,  $E_{f1}$ , and **In-plane Poisson's ratio of fiber**,  $\nu_{f12}$ . 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**,  $\sigma_{1D}$ . Also, enter the **Slope of in-plane fracture envelope, tension**,  $p_{t1}$ , and the **Slope of in-plane fracture envelope, compression**,  $p_{c1}$ . 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**  $\sigma_{ts}$ , **Compressive strengths**  $\sigma_{cs}$ , and **Shear strengths**  $\sigma_{ss}$ . All entries have three components, related to the principal axes of orthotropy. Also, enter the **Ultimate tensile strain in longitudinal direction**,  $\epsilon_{ts1}$ .
  - Enter the **In situ transverse tensile strength**,  $\sigma_{ts2}^i$ , and **In situ in-plane shear strength**,  $\sigma_{ss12}^i$ . For both parameters, the default is to compute the value from the data given above, 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 own expressions.
  - Enter **Young's modulus in longitudinal direction**,  $E_1$ , **Young's modulus in transverse direction**,  $E_2$ , **In-plane Poisson's ratio**,  $\nu_{12}$ , and **In-plane shear modulus**,  $G_{12}$ . For all four parameters, the default selection is **From parent**, indicating that the values are

taken from the linear elastic properties in the parent node. You can also take the values **From material**, or select **User defined** to enter own expressions.

- Enter the **Fracture plane angle under transverse compression**,  $\alpha_0$ . The default value is  $53^\circ$ . Also enter the **Fracture plane search angle**,  $\Delta\alpha$ . The default value is  $3^\circ$ . Under combined loading, the fracture plane angle will differ from  $\alpha_0$ , and a numerical search for the critical angle is performed in the range  $0 < \alpha < \alpha_0$ . The step in the search is  $\Delta\alpha$ .
- When **Failure Criterion** is **Tsai-Wu Anisotropic**, enter **Second rank tensor, Voigt notation**  $\mathbf{f}$ , and **Fourth rank tensor**  $\mathbf{F}$ . Enter the components of the tensors with respect to the directions of the coordinate system in the parent node.
- 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*

**Solid Mechanics>Linear Elastic Material>Variables>Safety**  
**Solid Mechanics>Nonlinear Elastic Material>Variables>Safety**  
**Shell>Linear Elastic Material>Variables>Safety**  
**Shell>Layered Linear Elastic Material>Variables>Safety**  
**Layered Shell>Linear Elastic Material>Variables>Safety**  
**Plate>Linear Elastic Material>Variables>Safety**  
**Membrane>Layered Linear Elastic Material>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**  
**Pipe Mechanics>Fluid and Pipe Materials>Variables>Safety**

### Ribbon

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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.

You can add the **Damping** subnode to the [Linear Elastic Material](#), the [Nonlinear Elastic Material](#), and the [Hyperelastic Material](#).

The following types of damping are available:

- [Rayleigh Damping](#)
- [Isotropic Loss Factor](#)
- [Anisotropic Loss Factor](#)
- [Orthotropic Loss Factor](#)
- [Viscous Damping](#)
- [Loss Factor](#)

The available damping models differs between various COMSOL products (see <https://www.comsol.com/products/specifications/>).



For a detailed description of the various damping models, see [Mechanical Damping and Losses](#).

---

The applicability of the different types of damping are summarized in [Table 4-4](#).

TABLE 4-4: AVAILABLE DAMPING TYPES

DAMPING TYPE	PHYSICS INTERFACES	MATERIAL MODELS	STUDY TYPES
<b>Rayleigh damping</b>	All	All	Time domain, frequency domain
<b>Isotropic loss factor</b>	All	All	Frequency domain

TABLE 4-4: AVAILABLE DAMPING TYPES

DAMPING TYPE	PHYSICS INTERFACES	MATERIAL MODELS	STUDY TYPES
<b>Anisotropic loss factor</b>	All, except Beam, Pipe Mechanics, and Truss	Linear Elastic	Frequency domain
<b>Orthotropic loss factor</b>	All, except Beam, Pipe Mechanics, and Truss	Orthotropic Linear Elastic	Frequency domain
<b>Viscous damping</b>	Solid Mechanics, Membrane, Multibody Dynamics	All	Time domain, frequency domain
<b>Viscous damping</b>	Shell	All	Time domain, frequency domain
<b>Loss factor</b>	Shell, Plate	Section Stiffness	Frequency domain

## 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.
- **Layered Linear Elastic Material** in the Shell interface. See the documentation for the [Damping](#) node in the [Shell and Plate](#) chapter.
- **Layered Linear Elastic Material** in the Membrane interface. See the documentation for the [Damping](#) node in the [Membrane](#) chapter.

## DAMPING SETTINGS

Select a **Damping type**, and enter settings depending on the type.

### *Rayleigh Damping*

In this damping model, the damping parameter  $\xi$  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*  $\alpha_{dM}$  and the *stiffness damping parameter*  $\beta_{dM}$ .

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, the **Mass damping parameter**  $\alpha_{dM}$  and the **Stiffness damping parameter**  $\beta_{dK}$ .

When **Damping ratios** is selected, enter two pairs of frequencies,  $f_1$  and  $f_2$ , and the corresponding damping ratios  $\zeta_1$  and  $\zeta_2$  at these frequencies. The Rayleigh damping parameters are computed as

$$\alpha_{dM} = 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{dK} = \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi(f_2^2 - f_1^2)}$$

#### *Isotropic Loss Factor*

The isotropic loss factor damping is described by the single isotropic loss factor  $\eta_s$ , 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  $\eta_s$ . The default is to take the value **From material**. For **User defined**, enter another value or expression.

#### *Anisotropic Loss Factor*

An elastic material is in general described by a symmetric 6-by-6 elasticity matrix  $D$ . The loss can be isotropic or anisotropic, and is described by either the isotropic loss factor  $\eta_s$  or by a symmetric anisotropic 6-by-6 loss factor matrix  $\eta_D$  or  $\eta_{DV_0}$ . 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  $\eta_D$  or  $\eta_{DV_0}$ . The default is to take the values **From material**. For



**User defined** enter the components of  $\eta_D$  or  $\eta_{DV_0}$  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  $\eta_D$  using the standard notation.

#### *Orthotropic Loss Factor*

This option is only available when **Orthotropic** is selected as the [Linear Elastic Material Solid model](#).

An orthotropic material is described by three Young's modulus components ( $E_x$ ,  $E_y$ , and  $E_z$ ) and three shear modulus components ( $G_{xy}$ ,  $G_{yz}$ , and  $G_{xz}$ ). For an orthotropic material, loss factors can be specified in three different ways:

- Isotropic, as described under [Isotropic Loss Factor](#).
- Anisotropic, as described under [Anisotropic Loss Factor](#).
- Orthotropic, described by three plus three orthotropic loss factors corresponding to the elastic moduli components for the orthotropic material. The orientations are the same as in the parent **Linear Elastic Material** node.

When **Orthotropic loss factor** is selected, use the **Loss factor for orthotropic Young's modulus** list to select the way to enter  $\eta_E$ . The default is to take the values **From material**. For **User defined** enter other values or expressions.

Use the **Loss factor for orthotropic shear modulus** list to select the way to enter  $\eta_G$  or  $\eta_{GV_0}$ . The default is to take the values **From material**. For **User defined**, enter other values or expressions.



The values for the shear modulus 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 **Orthotropic** or **Orthotropic, Voigt notation** property group for the material.

### Viscous Damping

With viscous damping, the material will get additional stresses proportional to the strain rate. Enter **Bulk viscosity**  $\eta_b$  and **Shear viscosity**  $\eta_v$  to model damping caused by volume change and deformation, respectively.

### 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**,  $\eta_{DA}$ ; **Loss factor for stiffness matrix DB**,  $\eta_{DB}$ ; **Loss factor for stiffness matrix DD**,  $\eta_{DD}$ ; and **Loss factor for stiffness matrix DAs**,  $\eta_{DA_s}$ .

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 modal 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**  
**Shell>Linear Elastic Material>Damping**  
**Shell>Layered Linear Elastic Material>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>Layered Linear Elastic Material>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**  
**Multibody Dynamics>Linear Elastic Material>Damping**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **Hyperelastic Material**, **Layered Linear Hyperelastic Material**, **Nonlinear Elastic Material**, **Elastoplastic Soil Material**, **Section Stiffness**, or **Fluid and Pipe Properties** 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 the [Magnetostrictive Material](#), either using loss factor material data for the stiffness, or in the form of Rayleigh proportional damping.

## DAMPING SETTINGS

Select a **Damping type** — **Loss factor for cE**, **Isotropic loss factor**, or **Rayleigh damping**.

#### *Loss Factor for cE (Piezoelectric Material)*

By default, the **Loss factor for elasticity matrix cE**  $\eta_{cE}$  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**. By default, the **Loss factor for compliance matrix sE**  $\eta_{cs}$  uses values **From material**. For **User defined** enter other values or expressions in the matrix.

#### *Loss Factor for cH (Magnetostrictive Material)*

By default, the **Loss factor for elasticity matrix cH**  $\eta_{cE}$  uses values **From material**. For **User defined** enter other values or expressions in the matrix

#### *Isotropic Loss Factor*

By default, the **Isotropic structural loss factor**  $\eta_s$  uses values **From material**. For **User defined** enter another value or expression.

#### *Rayleigh Damping*

Enter the **Mass damping parameter**  $\alpha_{dM}$  and the **Stiffness damping parameter**  $\beta_{dK}$ .

In this damping model, the damping parameter  $\xi$  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*  $\alpha_{dM}$  and the stiffness damping parameter  $\beta_{dM}$ . Note that the beta-damping is applied only to the mechanical part of the problem. To include the Rayleigh damping effect for the piezoelectric coupling terms, add a [Coupling Loss](#) subnode.



See also [Piezoelectric Losses](#).

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Piezoelectric Material>Mechanical Damping**

**Solid Mechanics>Magnetostrictive Material>Mechanical Damping**

**Layered Shell>Piezoelectric Material>Mechanical Damping**

### *Ribbon*

Physics tab with **Piezoelectric Material** or **Magnetostrictive 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.

#### **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  $\eta_e$  in a 3-by-6 matrix.
- For **Rayleigh damping** enter a **Stiffness damping parameter**  $\beta_{dC}$ .



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 electrical permittivity, or in the form of dielectric dispersion.

## DIELECTRIC LOSS SETTINGS

From the **Dielectric loss** list, select **Loss factor for  $\epsilon S$**  or **Dispersion**.



**Dispersion** option is only present when **Dielectric Loss** is used as a subnode to **Piezoelectric Material** in the Solid Mechanics interface.

---

For **Loss factor for  $\epsilon S$**  select a **Loss factor for electrical permittivity  $\epsilon S$** . Select **From material** to use the value from the material or select **User defined** to enter values or expressions for the loss factor in the associated fields. Select **Symmetric** to enter the components of  $\eta_{eS}$  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.

For **Dispersion** enter the **Relaxation time  $\tau_d$** , and the **Relative permittivity increment  $\Delta\epsilon_{rS}$**  in the associated fields. For the latter, you can select **Isotropic**, **Diagonal**, or **Symmetric** matrix input options.



See also [Piezoelectric Losses](#).

---

## 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.

## CONDUCTION CURRENT

By default, the **Electrical conductivity**  $\sigma_e$  for the media 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 electrical conductivity, and then enter values or expressions for the **Electrical conductivity**  $\sigma_e$  in the field or matrix.
- For **Linearized resistivity** the default **Reference temperature**  $T_0$ , and **Resistivity temperature coefficient**  $\alpha_r$ , and **Reference resistivity**  $\rho_0$  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 second Piola–Kirchhoff 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 <https://www.comsol.com/products/specifications/>).

### 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 Green–Lagrange strains, the deformation gradient, as well as all standard model inputs.

### GEOMETRIC NONLINEARITY

The use of an **External Stress-Strain Relation** will always force the study to be geometrically nonlinear. The default behavior is then to use a large strain formulation in all domains. There are, however, some cases when the use of a small strain formulation for a certain domain is needed. This will for example be the case if the material model you have implemented is formulated using engineering strains.

In such cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.



- [Modeling Geometric Nonlinearity](#)
- [Studies and Solvers](#) 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 non-isotropic materials where the normal to the surface is a principal direction. If this is not the case, select the **Transverse shear strains** check box to store state variables also for the two transverse shear strains.



## 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 Domain*

---

Add the **Rigid Domain** node and select one or more geometrical objects to make them a rigid body. **Rigid Domain** is a material model, with only one material property: the mass density. It can be used for

- Domains in the Solid Mechanics interface.
- 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 \(Rigid Domain\)](#)).

You can add functionality to the rigid domain through the following subnodes:

- [Fixed Constraint \(Rigid Domain\)](#) to fully constrain the rigid domain.
- [Prescribed Displacement/Rotation](#) to prescribe the displacement of individual degrees of freedom.
- [Applied Force \(Rigid Domain\)](#) to apply a force in given point.
- [Applied Moment \(Rigid Domain\)](#) to apply a moment.
- [Mass and Moment of Inertia \(Rigid Domain\)](#) to add extra mass and moment of inertia in a given point.
- [Spring Foundation \(Rigid Domain\)](#) to add a translational or rotational spring or damper in a given point.

The **Rigid Domain** node is only available with some COMSOL products (see <https://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 Domain** is used in the Layered Shell interface. See the documentation for the **Rigid Domain** node in the **Layered Shell** chapter.

## DENSITY

The default **Density**  $\rho$  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  $T_{\text{ref}}$  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 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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$



This section is not available in 2D axisymmetric components.

## 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**.



[Rigid Domain Model](#)



## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Material Models>Rigid Domain**

**Shell>Material Models>Rigid Domain**

**Layered Shell>Material Models>Rigid Domain**

**Beam>Material Models>Rigid Domain**

### *Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>Material Models>Rigid Domain**

Physics tab with **Shell** or **Layered Shell** selected:

**Boundaries>Material Models>Rigid Domain**

Physics tab with **Beam** selected:

**Edges>Material Models>Rigid Domain (3D)**

**Boundaries>Material Models>Rigid Domain (2D)**

### *Initial Values (Rigid Domain)*

The **Initial Values** node is automatically added as a default node to a [Rigid Domain](#) 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 Domain** 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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$

#### 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**  $\Omega$
- **Angle of rotation**  $\phi$ .
- **Angular velocity**  $\omega$  (3D components) and  $\partial\phi/\partial t$  (2D components)

#### LOCATION IN USER INTERFACE

This node is automatically added when the Rigid Domain node is created. It cannot be added or removed manually.

#### *Fixed Constraint (Rigid Domain)*

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 Domain](#).



The **Fixed Constraint** and **Prescribed Displacement/Rotation** subnodes cannot be combined for a Rigid Domain.

## REACTION FORCE SETTINGS

Select **Evaluate reaction forces** 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. Note that constraint groups only can be used when the constraints are pointwise, but not for weak constraints. This means that if **Evaluate reaction forces** is selected, any constraint group settings are ignored.

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Rigid Domain>Fixed Constraint**

**Shell>Rigid Domain>Fixed Constraint**

**Layered Shell>Rigid Domain>Fixed Constraint**

**Beam>Rigid Domain>Fixed Constraint**

**Pipe Mechanics>Rigid Domain>Fixed Constraint**

**Multibody Dynamics>Rigid Domain>Fixed Constraint**

### *Ribbon*

Physics tab with **Rigid Domain** 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 degrees of freedom of a Rigid Domain:

- 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 Domain.

### 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 Domain** 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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$

### 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) check boxes. Then enter a value or 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, enter a value or expression for the **Angle of rotation**  $\phi_0$ .



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** check boxes.
- For **Prescribed rotation** enter values or expressions in the table for the **Axis of rotation**  $\Omega$ . Then enter a value or expression for the **Angle of rotation**  $\phi_0$ .



- 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. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter. Note that constraint groups only can be used when the constraints are pointwise, but not for weak constraints. This means that if **Evaluate reaction forces** is selected, any constraint group settings are ignored.
- 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** 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 Domain>Prescribed Displacement/Rotation**

**Shell>Rigid Domain>Prescribed Displacement/Rotation**

**Layered Shell>Rigid Domain>Prescribed Displacement/Rotation**

**Beam>Rigid Domain>Prescribed Displacement/Rotation**

**Pipe Mechanics>Rigid Domain>Prescribed Displacement/Rotation**

**Multibody Dynamics>Rigid Domain>Prescribed Displacement/Rotation**

### *Ribbon*

Physics tab with **Rigid Domain** 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 Domain)*

---

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 any applicable coordinate systems that the model contains. 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 Domain** 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** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

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 F**. The direction coordinate names can vary depending on the selected coordinate system.



- 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 Domain>Applied Force**

**Shell>Rigid Domain>Applied Force**

**Layered Shell>Rigid Domain>Applied Force**

**Beam>Rigid Domain>Applied Force**

**Pipe Mechanics>Rigid Domain>Applied Force**

**Multibody Dynamics>Rigid Domain>Applied Force**

### *Ribbon*

Physics tab with **Rigid Domain** 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 Domain)*

---

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 any applicable coordinate systems that the model contains. 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,  $M_z$ .



- 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 Domain>Applied Moment**

**Shell>Rigid Domain>Applied Moment**

**Layered Shell>Rigid Domain>Applied Moment**

**Beam>Rigid Domain>Applied Moment**

**Pipe Mechanics>Rigid Domain>Applied Moment**

**Multibody Dynamics>Rigid Domain>Applied Moment**

### *Ribbon*

Physics tab with **Rigid Domain** node selected in the model tree:

**Attributes>Applied Moment**

### *Mass and Moment of Inertia (Rigid Domain)*

---

Use the **Mass and Moment of Inertia** subnode to add an effect of 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 Domain** 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**,  $\mathbf{X}_m$ , in the table.

Select the **Offset** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_m = \mathbf{X}_{m,\text{input}} + \mathbf{X}_{\text{offset}}$$

#### 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 expression for  $I_z$ .

#### LOCATION IN USER INTERFACE

##### *Context Menus*

**Solid Mechanics>Rigid Domain>Mass and Moment of Inertia**

**Shell>Rigid Domain>Mass and Moment of Inertia**

**Layered Shell>Rigid Domain>Mass and Moment of Inertia**

**Beam>Rigid Domain>Mass and Moment of Inertia**

**Pipe Mechanics>Rigid Domain>Mass and Moment of Inertia**

**Multibody Dynamics>Rigid Domain>Mass and Moment of Inertia**

##### *Ribbon*

Physics tab with **Rigid Domain** 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 \(Rigid Domain\)](#).

#### 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 \(Rigid Domain\)](#).

## 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 \(Rigid Domain\)](#).

## LOCATION IN USER INTERFACE

These nodes are automatically added when **Centroid of selected entities** is selected in the parent [Mass and Moment of Inertia \(Rigid Domain\)](#) node. They cannot be added or removed manually.

### *Spring Foundation (Rigid Domain)*

---

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 Domain** 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 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** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_p = \mathbf{X}_{p,\text{input}} + \mathbf{X}_{\text{offset}}$$

### 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  $\eta_f$  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 Domain>Spring Foundation**

**Shell>Rigid Domain>Spring Foundation**

**Layered Shell>Rigid Domain>Spring Foundation**

**Beam>Rigid Domain>Spring Foundation**

**Pipe Mechanics>Rigid Domain>Spring Foundation**

**Multibody Dynamics>Rigid Domain>Spring Foundation**

### *Ribbon*

Physics tab with **Rigid Domain** 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 which do not have any boundary conditions applied.

In each physics interface, the **Free** node applies to a geometric entity which 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>Free**

**Shell>Free**

**Plate>Free**

**Layered Shell>Free**

**Membrane>Free**

**Beam>Free**

**Pipe Mechanics>Free**

**Truss>Free**

**Multibody Dynamics>Free**

### *Ribbon*

Physics tab with a physics interface selected:

**Boundaries>Solid Mechanics>Free**

**Boundaries>Plate>Free**

**Boundaries>Multibody Dynamics>Free**

**Edges>Shell>Free**

**Edges>Layered Shell>Free**

**Edges>Membrane>Free**

**Points>Beam>Free**

**Points>Pipe Mechanics>Free**

**Points>Truss>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).

If a displacement is prescribed in one direction, this leaves the solid free to deform in the other directions.

You can also define more general displacements as a linear combination of the displacements in each direction.



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 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 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 which has fixed spatial directions.

### **PRESCRIBED DISPLACEMENT**

Define the prescribed displacements using a **Standard notation** or a **General notation**.

### Standard Notation

To define the displacements individually, click the **Standard notation** button.

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and for 3D components, **Prescribed in z direction** check boxes. Then enter a value or expression for  $u_0$ ,  $v_0$ , and for 3D components,  $w_0$ . For 2D axisymmetric components, select one or both of the **Prescribed in r direction** and **Prescribed in z direction** check boxes. Then enter a value or expression for  $u_0$  and  $w_0$ .

### General Notation

Click the **General notation** to 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

$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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 box.



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*

**Solid Mechanics>Domain Constraints>Prescribed Displacement**

**Solid Mechanics>Prescribed Displacement** (Boundary)

**Solid Mechanics>Edges>Prescribed Displacement**

**Solid Mechanics>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)

**Multibody Dynamics>Domain Constraints>Prescribed Displacement**

**Multibody Dynamics>Prescribed Displacement** (Boundary)

**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 **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 **Multibody Dynamics** selected:

**Domains>Domain Constraints>Prescribed Displacement**  
**Boundaries>Multibody Dynamics>Prescribed Displacement**  
**Edges>Multibody Dynamics>Prescribed Displacement**  
**Points>Multibody Dynamics>Prescribed Displacement**

### *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 <https://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 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 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 for 3D components, **Prescribed in z direction** check boxes. Then enter a value or expression for  $v_x$ ,  $v_y$ , and for 3D components,  $v_z$ . For 2D axisymmetric components, select one or both of the **Prescribed in r direction** and **Prescribed in z direction** check boxes. Then enter a value or expression for  $v_r$  and  $v_z$ .

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)

### *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** selected:

**Edges>Line Constraints>Prescribed Velocity**  
**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. 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 <https://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 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 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 for 3D components, **Prescribed in z direction** check boxes. Then enter a value or expression for  $a_x$ ,  $a_y$ , and for 3D components,  $a_z$ . For 2D axisymmetric components, select one or both of the **Prescribed in r direction** and **Prescribed in z direction** check boxes. Then enter a value or expression for  $a_r$  and  $a_z$ .

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)**

### *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** selected:

**Edges>Line Constraints>Prescribed Acceleration**  
**Points>More Constraints>Prescribed Velocity**

### *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 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 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 box.



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**

**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** 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**

### *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 temperature of the surroundings. This makes it possible to reduce the stresses caused by such boundary conditions.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

The **Thermal Expansion** subnode is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

## **SHELL PROPERTIES**



This section is only present in the 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** check box 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 **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.

- The **Coefficient of thermal expansion**  $\alpha$ . As a 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.

For **User defined** select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.

- A value or expression for the **Volume reference temperature**  $T_{\text{ref}}$  that 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*

**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.

## ROLLER CONSTRAINT

The default roller constraint acts along a normal to the selected boundaries which is computed numerically. For a certain mesh node, 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**.


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.

For **Cylinder**, select **Axis type**— **X-axis**, **Y-axis**, **Z-axis**, or **User defined** to describe the orientation of the cylinder. Enter the location of a **Point on axis**,  $\mathbf{X}_c$ , in order fix the location of the cylinder. When **User defined** is used, enter the vector  $\mathbf{e}_s$ , pointing along the axis of the cylinder. The constraints will act radially from the cylinder axis. If the analysis is geometrically nonlinear, each mesh node 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 node 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 box.



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

- [Rigid Connector Theory](#)

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*

**Solid Mechanics>Roller**

**Layered Shell>Face and Domain Constraints>Interfaces>Roller, Interface**



**Layered Shell>Roller**  
**Multibody Dynamics>Roller**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Boundaries>Solid Mechanics>Roller**

Physics tab with **Layered Shell** selected:

**Boundaries>Interfaces>Roller, Interface Edges>Layered Shell>Roller**

Physics tab with **Multibody Dynamics** selected:

**Boundaries>Multibody Dynamics>Roller**

*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 <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES

---



This section is only present in the 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 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.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 box.



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*

**Solid Mechanics>More Constraints>Symmetry**

**Layered Shell>More Constraints>Symmetry**

**Membrane>More Constraints>Symmetry**

### *Ribbon*

Physics tab with **Solid Mechanics** 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. A symmetry condition is free in the plane and fixed in the out-of-plane direction.

The **Symmetry** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

## 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 box.



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*

**Solid Mechanics>More Constraints>Symmetry Plane  
Membrane>More Constraints>Symmetry Plane**

### *Ribbon*

Physics tab with **Solid Mechanics** 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 <https://www.comsol.com/products/specifications/>).



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 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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry Constraints</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>More Constraints>Antisymmetry**

**Layered Shell>Antisymmetry**

**Membrane>More Constraints>Antisymmetry**

### *Ribbon*

Physics tab with **Solid Mechanics** 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 a minimum number of constraints required to suppress any rigid body modes. The constraints are selected so that there will be no reaction forces if the external loads are self-equilibrating.

The constraint will, depending on physics interface and geometrical dimension, be applied to one, two, or three points as needed. These points are automatically picked from the selected geometrical objects.

This feature must not be combined with any other types of constraints, such as prescribed displacements or symmetry conditions.

If you need to constrain several objects which are not physically connected, you must add as many **Rigid Motion Suppression** nodes as there are disconnected objects.

## SHELL PROPERTIES.

	This section is only present in the in the Layered Shell interface, where it is described in the documentation for the <a href="#">Rigid Motion Suppression</a> node.
	In the <i>COMSOL Multiphysics Reference Manual</i> : <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<ul style="list-style-type: none"><li>• <a href="#">Rigid Body Motion</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</li><li>• <a href="#">Rigid Motion Suppression Boundary Condition</a> in the <a href="#">Structural Mechanics Theory</a> chapter.</li></ul>
	You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> 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**  
**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 in the Layered Shell interface, where it is described in the documentation for the [Body Load](#) node.

---

## **FORCE**

Select a **Load type** — **Force per unit volume**, **Total force**, or for 2D components, **Force per unit area**.



Then enter values or expressions for the components in the matrix based on the selection and the space dimension.



- 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 unit area** is selected, the body load as force per unit volume is then the value of  $F$  divided by the thickness.

TABLE 4-5:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit volume	$F_V$	N/m <sup>3</sup>	domains	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per unit area	$F_A$	N/m <sup>2</sup>	domains	2D (x, y)
Total force	$F_{tot}$	N	domains	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)



- 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**

**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** 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 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 which belong to the selection are automatically included.

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). The action of gravity can also be seen as a linearly accelerated frame of reference.

The **Gravity** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES



This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Gravity](#) node.

---

## 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 gravitational forces.

## GRAVITY

Enter the components of the **Gravity**  $\mathbf{g}$ . The default value is  $g\_const$ , which is the physical constant having the value  $9.8066 \text{ m/s}^2$ .

For 2D components, the default is that the gravity acts in the negative  $y$  direction. For 3D and 2D axisymmetric components, the default is that the gravity acts in the negative  $z$  direction.



## Rotating Frame and Gravity



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 **Gravity**.



- 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>Volume Forces>Gravity**

**Shell>Face and Volume Loads>Gravity**

**Layered Shell>Face and Volume Loads>Gravity**

**Plate>Face and Volume Loads>Gravity**

**Membrane>Face and Volume Loads>Gravity**

**Beam>Line and Volume Loads>Gravity**

**Truss>Line and Volume Loads>Gravity**

**Multibody Dynamics>Volume Forces>Gravity**

### *Ribbon*

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

**Domains>Volume Forces>Gravity**

Physics tab with **Shell**, **Layered Shell**, or **Membrane** selected:

**Boundaries>Face and Volume Loads>Gravity**

Physics tab with **Plate** selected:

**Domains>Face and Volume Loads>Gravity**

Physics tab with **Beam** or **Truss** selected:

**Edges>Line and Volume Loads>Gravity**

### *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 which 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 <https://www.comsol.com/products/specifications/>).

### **SHELL PROPERTIES**



This section is only present in the 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 direction** — **Counterclockwise** or **Clockwise**. The rotational direction does not make any difference for a centrifugal force.

Select a **Rotational frequency** — **Angular velocity magnitude**, **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 magnitude** enter a value for the angular velocity magnitude  $\Omega$ .
- For **Revolutions per time**, enter an RPM value.
- For **Rigid body**, enter an **Applied moment**,  $M$ , and an **Initial angular velocity**,  $\Omega_0$ . 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.

$$I_{ax} \frac{\partial \Omega}{\partial t} = M$$
$$I_{ax} = \int \rho r_p^2 dV$$

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**  $\alpha$  in radians as function of time (the variable  $t$ ).

In the case of a geometrically nonlinear analysis, the displacements computed in the Solid Mechanics interface are used to defined 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 non-constant, 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  $\alpha_0 + \Omega t$ .

#### FRAME ACCELERATION EFFECT

For 3D and 2D components, use the **Centrifugal force**, **Coriolis force**, or **Euler force** check boxes 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** check box 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** check box is only available if **Centrifugal force** or **Euler force** is selected.



[Rotating Frame and Gravity](#)



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**.



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**

**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**  
**Multibody Dynamics>Volume Forces>Rotating Frame**

### *Ribbon*

Physics tab with **Solid Mechanics** 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** or **Truss** selected:

**Edges>Line and Volume Loads>Rotating Frame**

### *Boundary Load*

---

Use a **Boundary Load** to apply tractions or pressure to boundaries.

## **SHELL PROPERTIES**



This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Boundary Load](#) node.

---

## **FORCE**

Select a **Load type** — **Force per unit area**, **Pressure**, **Total force**, or for 2D components, **Force per unit length**. Then enter values or expressions for the components in the matrix based on the selection and the space dimension.

- For **Force per unit 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 unit 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 toward the solid. In a geometrically nonlinear analysis, the current surface normal and area are used.



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 4-6:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit area	$F_A$	N/m <sup>2</sup>	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per unit length	$F_L$	N/m	boundaries	2D (x, y)
Total force	$F_{tot}$	N	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Pressure	$p$	Pa	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)



- 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**



**Layered Shell>Boundary Load**  
**Multibody Dynamics>Boundary Load**

*Ribbon*

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

**Boundaries>Solid Mechanics>Body Load**  
**Boundaries>Multibody Dynamics>Body Load**

Physics tab with **Layered Shell** selected:

**Edges>Layered Shell>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 in the Layered Shell interface, where it is described in the documentation for the [Edge Load](#) node.

---

#### FORCE

Select a **Load type** — **Force per unit length** or **Total force**. Then enter values or expressions for the components in the matrix based on the selection:

- The load per unit length  $\mathbf{F}_L$ .
- The total force  $\mathbf{F}_{\text{tot}}$ . COMSOL Multiphysics then divides the total force by the volume where the load is active.



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.

---



- 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**

**Layered Shell>Edge Load**

**Multibody Dynamics>Edges>Edge Load**

### *Ribbon*

Physics tab with **Solid Mechanics** 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 in the Layered Shell interface, where it is described in the documentation for the [Point Load](#) node.

## FORCE

Enter values or expressions for the components of the point load  $\mathbf{F}_p$ .



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.



- 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**

**Layered Shell>Points>>Interfaces>Point Load**

**Truss>Point Load**

**Multibody Dynamics>Points>Point Load**

### *Ribbon*

Physics tab with **Solid Mechanics**, **Multibody 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**

### *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.

## FORCE

Select the **Load type** — **Force per unit length** or **Total force**. Enter values or expressions for  $\mathbf{F}_L$  or  $\mathbf{F}_{\text{tot},f}$ .



- 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 unit length values, multiplied by  $2\pi r$ , where  $r$  is the radius of the point.
- 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**

*Point Load (on Axis)*

A **Point Load (on Axis)** node can be added to points located at  $R = 0$  in 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.
- 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)**

**Shell>Point Load (on Axis)**

**Membrane>Point Load (on Axis)**

### *Ribbon*

Physics tab with **Solid Mechanics** 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)**

### *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 Elastic 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 Elastic 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 <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES



This section is only present in the 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 in the Layered Shell interface, where it is described in the documentation for the [Spring Foundation, Interface](#) node.

## 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 unit volume	$k_V$	$N/(m \cdot m^3)$	domains	3D, 2D, and 2D axisymmetric
Total spring constant	$k_{tot}$	$N/m$	domains, edges	3D, 2D, and 2D axisymmetric
Spring constant per unit area	$k_A$	$N/(m \cdot m)^2$	boundaries	3D, 2D
Spring constant per unit length	$k_L$	$N/(m \cdot m)$	edges, boundaries (2D)	3D, 2D
Spring constant	$k_P$	$N/m$	points	3D, 2D, and 2D axisymmetric
Force per volume as function of extension	$F_V$	$N/m^3$	domains	3D, 2D, and 2D axisymmetric
Total force as function of extension	$F_{tot}$	$N$	domains, boundaries, edges	3D, 2D, and 2D axisymmetric
Force per area as function of extension	$F_A$	$N/m^2$	domains, boundaries	3D, 2D
Force per length as function of extension	$F_L$	$N/m$	edges	3D
Force as function of extension	$F_P$	$N$	points	3D, 2D, and 2D 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	$\nu =$	$\nu$	$\frac{E}{2G} - 1$	$\frac{1}{2}\left(1 - \frac{3G}{3K+G}\right)$
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 interface. 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**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring**  $\eta_s$  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**  $\eta_k$  or  $\eta_f$  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 unit volume	$d_V$	$N \cdot s / (m \cdot m^3)$	domains, boundaries (2D)	3D, 2D
Damping constant per unit area	$d_A$	$N \cdot s / (m \cdot m^2)$	domains, boundaries	3D, 2D, and 2D axisymmetric
Total damping constant	$d_{tot}$	$N \cdot s / m$	domains, boundaries, edges, points	3D, 2D, and 2D axisymmetric
Damping constant per unit length	$d_L$	$N \cdot s / (m \cdot m)$	edges	3D

### 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.



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>Spring Foundation** (Domain, Boundary)

**Solid Mechanics>Edges>Spring Foundation**

**Solid Mechanics>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**  
**Multibody Dynamics>Mass, Spring, and Damper>Spring Foundation** (Domain)  
**Multibody Dynamics>Spring Foundation** (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 **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**, or **Truss** 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>Multibody Dynamics>Spring Foundation**

**Edges>Multibody Dynamics>Spring Foundation**

**Points>Multibody Dynamics>Spring Foundation**

### *Thin Elastic Layer*

---

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”.

The **Thin Elastic Layer** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

## **SHELL PROPERTIES**

---



This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Thin Elastic Layer](#) node.

---

## **INTERFACE SELECTION**

---



This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Thin Elastic Layer, 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. Ctrl-click to deselect.

## 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	$k_{\text{tot}}$	N/m	3D, 2D, and 2D axisymmetric
Spring constant per unit area	$k_A$	$\text{N}/(\text{m}\cdot\text{m})^2$	3D, 2D, and 2D axisymmetric
Spring constant per unit length	$k_L$	$\text{N}/(\text{m}\cdot\text{m})$	2D
Total force as function of extension	$F_{\text{tot}}$	N	3D, 2D, and 2D axisymmetric

TABLE 4-10: SPRING TYPES FOR THE THIN ELASTIC LAYER FEATURE

SPRING TYPE	VARIABLE	SI UNITS	SPACE DIMENSION
Force per area as function of extension	$F_A$	N/m <sup>2</sup>	3D, 2D, and 2D axisymmetric
Force per length as function of extension	$F_L$	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**  $\rho_A$  of the thin elastic layer.
- For **Spring constant per unit length** and **Force per length as function of extension**, enter the **Mass per unit length**  $\rho_L$  of the thin elastic layer.
- For **Use material data**, enter the **Density**  $\rho$  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**  $\eta_s$  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**  $\eta_k$  or  $\eta_f$  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	$d_A$	N·s/(m·m <sup>2</sup> )	3D, 2D, and 2D axisymmetric
Total damping constant	$d_{tot}$	N·s/m	3D, 2D, and 2D axisymmetric
Damping constant per unit length	$d_L$	N·s/(m·m)	2D



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**

**Solid Mechanics>Pairs>Thin Elastic Layer**

**Layered Shell>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**

Physics tab with **Layered Shell** selected:

**Edges>Mass, Spring, and Damper>Thin Elastic Layer**

## Predeformation

---

Use the **Predeformation** subnode to specify that the elastic forces in [Spring Foundation](#) or [Thin Elastic Layer](#) 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 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**  $u_0$ .

---



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 Elastic Layer>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**

**Multibody Dynamics>Spring Foundation>Predeformation**


### *Ribbon*

Physics tab with **Spring Foundation**, **Spring Foundation**, **Interface**, or **Thin Elastic Layer** node selected in the model tree:

#### **Attributes>Predeformation**

### *Spring-Damper*

---

Use a **Spring-Damper** () for 2D and 3D 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 by 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.

#### **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 Domain**
- **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 entered 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**,  $\Theta_d$ .



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 ( $\Theta_s$  and  $\Theta_d$ ) 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 if they have rotational degrees of freedom or not, a connection to an attachment or rigid body type object will provide a rotation.



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** check box 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**.

### *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>.d1, for example solid.spd1.d1.

To add viscous damping in a dynamic analysis, enter a value or 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 which 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 which 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.

#### 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  $\Delta l_0$ . 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** check box. Then, enter a **Deactivation expression**  $i_{\text{dac}}$ . 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** check box 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.d1>0.12[m]`, and select the **Permanently deactivate** check box. If the check box 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**

#### *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**

### *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.

### *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 <https://www.comsol.com/products/specifications/>).

## SHELL PROPERTIES



This section is only present in the 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 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 unit volume	$\rho_V$	kg/m <sup>3</sup>	domains, boundaries
Mass per unit area	$\rho_A$	kg/m <sup>2</sup>	domains, boundaries, edges
Mass per unit length	$\rho_L$	kg/m	edges, points
Total mass	<b>m</b>	kg	domains, boundaries, edges

## FRAME ACCELERATION FORCES

Select the **Exclude contribution** check box to switch off the loads that can be caused by the added mass when the frame is accelerated when using a [Gravity](#) or [Rotating Frame](#) feature. The setting will also determine whether the node will contribute when computing mass properties; see [Computing Mass Properties](#).

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Mass, Spring, and Damper>Added Mass** (Domain, Boundary)

**Solid Mechanics>Edges>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)

**Multibody Dynamics>Mass, Spring, and Damper>Added Mass** (Domain)

**Multibody Dynamics>Added Mass** (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 **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** or **Truss** selected:

**Edges>Mass, Spring, and Damper>Added Mass**

Physics tab with **Multibody Dynamics** selected:

**Domains>Mass, Spring, and Damper>Added Mass**

**Boundaries>Multibody Dynamics>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 modeled structure is typically a unit cell of a repetitive structure.
- The **Cyclic symmetry** is a special case of a Floquet condition, intended for structures which consist of a number of sectors which are identical when rotated around a common axis, like in a fan.

The **Floquet periodicity** and **Cyclic symmetry** options are available only with some COMSOL products (see <https://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

Select both the source and destination boundaries.

The software automatically identifies the boundaries as either source boundaries or destination boundaries. This works fine for cases like opposing parallel boundaries. In more general cases, use the **Destination Selection** subnode to specify the boundaries which constitute the destination. By default, this node contains the selection that COMSOL Multiphysics has identified.

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.

## 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;  $\mathbf{u}(\mathbf{x}_d) = \mathbf{u}(\mathbf{x}_s)$ . 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.
- For **Antiperiodicity** the displacements on the destination are set equal to the displacements on the source with the sign reversed;  $\mathbf{u}(\mathbf{x}_d) = -\mathbf{u}(\mathbf{x}_s)$ . 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.
- 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** 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  $\theta_S$ . 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 check box 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 box.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)

### ORIENTATION OF SOURCE

For information about this section, see [Orientation of 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**

**Solid Mechanics>Connections>Periodic Condition>Destination Selection**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Boundaries>Connections>Periodic Condition**

Physics tab with **Periodic Condition** node selected in the model tree:

**Attributes>Destination Selection**

*Cell Periodicity*

---

Use the **Cell Periodicity** node to model a unit cell (*representative volume element*, *RVE*) representing a large repetitive structure. Periodic boundary conditions will be used on the outer boundaries of this unit cell.

From the **Cell Periodicity** node, is possible to generate the elasticity matrix for the equivalent homogenized material.

If more than one **Cell Periodicity** node are used, they must have disjoint selections. It is thus possible to evaluate more than one RVE in the same study.

**PERIODICITY TYPE**

Select a **Periodicity type** — **Free expansion**, **Average strain**, **Average stress**, or **Mixed**.

With the **Free expansion** periodicity type, the unit cell is allowed to expand freely in a periodic manner. This option is useful to determine the thermal expansion or hygroscopic swelling coefficients of a heterogeneous unit cell. Select **Calculate average properties** — **None**, **Coefficient of thermal expansion**, or **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** periodicity type.

---

The **Average strain** periodicity type makes it possible to derive homogenized elastic properties of media with periodic structures, such as a perforated plates, porous media or composites structures. In this case, you have to examine six load cases in 3D. Enter values or expressions for the components of the **Average strain** tensor  $\epsilon_{\text{avg}}$ . In a geometrically nonlinear analysis, the average strains are interpreted as Green–Lagrange strains.

Select **Calculate average properties** — **None**, **Elasticity matrix**, **Standard (XX, YY, ZZ, XY, YZ, XZ)**, or **Elasticity matrix, Voigt (XX, YY, ZZ, YZ, XZ, XY)**.

The **Average stress** periodicity type makes it possible to derive the homogenized compliance matrix of media with periodic structures. Enter values or expressions for

the components of the **Average stress** tensor  $\sigma_{\text{avg}}$ . In a geometrically nonlinear analysis, the stresses are interpreted as Second Piola–Kirchhoff stresses.

Select **Calculate average properties** — **None**, **Compliance matrix**, **Standard (XX, YY, ZZ, XY, YZ, XZ)**, or **Compliance matrix, Voigt (XX, YY, ZZ, YZ, XZ, XY)**.



In 2D, use a **Generalized plane strain** approximation to calculate averaged properties with the **Average strain** or **Average stress** periodicity types.

With the **Mixed** periodicity type, enter either the average strain or average stress tensor components.



When using a **Plane stress** or a **Plane strain** 2D approximation with the **Mixed** periodicity type, 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.

For all values of **Periodicity type**, select the **Cell volume** — **Solid**, **Void volume fraction**, or **User defined**.

- When **Solid** is selected, the volume of the RVE is computed from the domains selected in the **Cell Periodicity** node.
- The **Void volume fraction** option can be used to scale the RVE volume when there are voids inside the RVE that are not selected as domains. Enter the void volume fraction,  $f$ . This is the fraction of the total RVE that is occupied by voids. The volume of the RVE is computed as


$$V_{\text{RVE}} = \frac{V_{\text{solid}}}{1-f}$$



- By using the **User defined** option, you can enter the volume of the RVE,  $V$ , explicitly.

The variable `<item>.vol` contains the volume of the RVE used when computing average strains and stresses.

#### *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 type** section toolbar, there is an icon **Study and Material Generation** () . It has a list with two entries:

- **Create Load Groups and Study** (). This option can be selected when **Periodicity Type** is **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 **Periodicity Type** 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** (). This option can be selected when **Periodicity Type** is **Average strain** and **Calculate average properties** not set to **None**. When you select it, a new material will be created under **Global->Materials**. It contains the, in general anisotropic, elasticity matrix. The name of this material is **Homogeneous Material**.



For this type of analysis to work correctly, it is important that you do not edit the generated nodes manually. By clicking the **Create** 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.
- [Periodic Cell Theory](#) in the Structural Mechanics Theory chapter.
- [Materials and Solid Mechanics Material Properties](#) in the Materials chapter in the *COMSOL Multiphysics Reference Manual*.



- *Micromechanical Model of a Composite*: Application Library path **Structural\_Mechanics\_Module/Material\_Models/micromechanical\_model\_of\_a\_composite**
- *Micromechanics and Stress Analysis 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 box.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Cell Periodicity**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>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 <https://www.comsol.com/products/specifications/>).

## DAMPING

Select a **Damping type** — **P and S waves** or **User defined**. For **User defined** enter values or expressions for the **Mechanical impedance**  $d_i$ . The defaults for all values are  $0.5 \cdot \text{solid.rho} \cdot (\text{solid.cp} + \text{solid.cs})$ .



Low-Reflecting Boundary Condition

---

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Mass, Spring, and Damper>Low-Reflecting Boundary**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Boundaries >Mass, Spring, and Damper>Low-Reflecting Boundary**



## Thin-Film Damping

---



The MEMS Module also includes the Thin-Film Flow interface, which, when appropriately coupled with the Solid Mechanics interface, is equivalent to using the Thin-Film Damping boundary condition.

---

Use the **Thin-Film Damping** node to apply boundary loads to the surface of a resonator that result from squeeze-film or slide-film damping. Squeeze-film damping occurs when a thin film of gas is “squeezed” between two parallel plates, one of which is in motion normal to its surface. Slide-film damping occurs in similar situations except that the direction of motion of the moving plate is tangential to its surface. 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. The thin-film damping boundary condition can accommodate general displacements of the moving plates specified by means of the **Fluid-Film Properties** node.

The **Thin-Film Damping** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

The **Fluid-Film Properties** node and the **Border** node are added by default. These additional subnodes are available: **Fluid-Film Properties**, **Border**, **Inlet**, **Outlet**, **Wall**, and **Symmetry**.

---



The [Fluid-Film Properties](#), [Inlet](#), [Outlet](#), [Border](#), [Wall](#), and [Symmetry](#) nodes are all described for [The Thin-Film Flow Interfaces](#) in the *MEMS Module User's Guide*.

Also see [Theory for the Thin-Film Flow Interfaces](#) in the *MEMS Module User's Guide* as the **Thin-Film Damping** boundary condition is equivalent to an appropriately coupled Thin-Film Flow interface. Therefore, its theory is described in full for that physics interface.

---

### LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Mass, Spring, and Damper>Thin-Film Damping**

## Ribbon

Physics tab with **Solid Mechanics** selected:

**Boundaries>Mass, Spring, and Damper>Thin-Film Damping**

### *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 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 (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.

Similar rigid connector features are available also in the Beam and Shell interfaces. Rigid connectors from Beam, Shell, and Solid Mechanics interfaces can be attached to each other by coupling their translational or rotational degrees of freedom manually.

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 <https://www.comsol.com/products/specifications/>).

#### SHELL PROPERTIES



This section is only present in the 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 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 may 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** check box in the **Advanced** section.

---



### **PAIR SELECTION**

If this node is selected from the **Pairs** menu, choose the pair to use. An identity pair has to be created first. The rigid connector applies to the common part of the boundaries, and makes the parts behave as if there were an infinitely stiff layer between them.

### **COORDINATE SYSTEM SELECTION**

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any applicable coordinate systems that the model includes. 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.

### **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, since 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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $X_{\text{offset}}$ .

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 at the center of rotation for each space direction, select one or several of the available check boxes 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**  $u_{0x}$
- **Prescribed in y direction**  $u_{0y}$
- *For 3D components:* **Prescribed in z direction**  $u_{0z}$

## PREScribed ROTATION AT CENTER OF ROTATION

Specify the rotation at the center of rotation. Select from the **By** list: **Free**, **Constrained rotation**, or **Prescribed rotation at center of rotation**.



For 2D components, the **Constrained rotation** and **Prescribed rotation at center of 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 check boxes 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 at Center of Rotation*

For **Prescribed rotation at center of rotation** enter an **Angle of rotation**  $\phi_0$ . For 3D components also enter an **Axis of rotation**  $\Omega$  for the **x**, **y**, and **z** coordinates.

## 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.

## 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

This section is present when the rigid connector is added at edge or point level. In these cases, checks for selections that would result in a singularity are performed as a default. If these checks give false positives, you can turn them off by clearing the **Include consistency checks** check box. 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.



- 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. Note that constraint groups only can be used when the constraints are pointwise, but not for weak constraints. This means that if **Evaluate reaction forces** is selected, any constraint group settings are ignored.
- 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 Theory](#)
- [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>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>Multibody Dynamics>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 \(Rigid Domain\)](#)
- [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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

### SHELL PROPERTIES

---



This section is only present in the 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 **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.
- The **Coefficient of thermal expansion**  $\alpha$ . As a default, values **From material** are used. This requires that a material has been assigned to the boundaries of the rigid connector selection.  
For **User defined** select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.
- A value or expression for the **Volume reference Temperature**  $T_{\text{ref}}$  which is the temperature at which there are no thermal displacements in the rigid connector.



- [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 any applicable coordinate systems that the model contains. Select a **Coordinate system** for specifying the directions of the force.

### 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** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

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.



- 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 any applicable coordinate systems that the model contains. 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,  **$M_z$** .



- 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**,  **$\mathbf{X}_m$** , in the table.

Select the **Offset** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  **$\mathbf{X}_{\text{offset}}$** .

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_m = \mathbf{X}_{m,\text{input}} + \mathbf{X}_{\text{offset}}$$

### 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 expression for  $I_z$ .

### LOCATION IN USER INTERFACE

#### *Context Menus*

**Solid Mechanics>Rigid Connector>Mass and Moment of Inertia**

**Shell>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)**,  $\mathbf{X}_p$ , in the table.



Select the **Offset** check box to add an optional offset vector to the definition of the location. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_P = \mathbf{X}_{P,\text{input}} + \mathbf{X}_{\text{offset}}$$

### 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.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  $\eta_f$  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 **Attachment** node is available with either the Multibody Dynamics Module or the Rotordynamics Module. The use of attachments is discussed in the [Attachments](#) section in the *Multibody Dynamics Module User's Guide*.

---

### **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



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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

## 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 **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.
- The **Coefficient of thermal expansion**  $\alpha$ . As a default, values **From material** are used. This requires that a material has been assigned to the boundaries of the attachment selection.

For **User defined** select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.

- A value or expression for the **Volume reference Temperature**  $T_{\text{ref}}$  which is the temperature at which there are no thermal displacements in the attachment.



- [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**



The **Beam Connection** node is obsolete, and has been superseded by the [Solid-Beam Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Beam Connection** node will be removed in future versions.

---

In 2D, a solid can be connected to a beam by adding a **Beam Connection** node in the Solid Mechanics interface and a [Solid Connection](#) node in the Beam interface. The connection can either be from a point on the beam to a boundary on the solid, or between two boundaries. The first case is intended for modeling a transition from a beam to a solid where beam theory assumptions are valid on both sides of the connection. The second case is for adding a beam on top of a solid as stiffener or cladding.

The **Beam Connection** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

---



- 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](#)
- 

#### BEAM CONNECTION

Select a **Connected entity** — **Beam point** or **Beam edge**.

When **Beam point** is selected, select the name of the corresponding **Solid Connection** node defined at the point level in the Beam interface to specify the connected parts.

Select an option from the **Connected area defined by** list: **Section height**, **Selected boundaries**, or **Distance from beam axis**. This parameter determines how much of the selected solid boundaries 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 both directions 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.

For **Beam edge** select the name of the corresponding **Solid Connection** node defined at the boundary level in the Beam interface to specify the connected parts.

Select a **Distance evaluation: Section height, Geometrical distance, or User defined**. This parameter determines how the coupling treats the determination of the connection distance. 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.

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Connections>Beam Connection**

### *Ribbon*

Physics tab with **Solid Mechanics** selected:

**Boundaries>Connections>Beam Connection**

### *Shell Connection*

---



The **Shell Connection** node is obsolete, and has been superseded by the [Solid-Thin Structure Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Shell Connection** node will be removed in future versions.

In 3D, a solid can be connected to a shell by adding a **Shell Connection** node in the Solid Mechanics interface and a [Solid Connection](#) node in the Shell interface. The connection can either be from an edge of the shell to a boundary on the solid, or between two boundaries. The first case is intended for modeling a transition from a shell to a solid where shell assumptions are valid on both sides of the connection. The second case is for adding a shell on top of a solid.

The **Shell Connection** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).



- 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](#)

## SHELL CONNECTION

Select a **Connected entity** — **Shell edge** or **Shell boundary**.

When **Shell edge** is selected, select the name of the corresponding **Solid Connection** node defined at the edge level in the Shell interface to specify the connected parts.

Select an option from the **Connected area defined by** list: **Shell thickness**, **Selected boundaries**, or **Distance from shell midsurface**. This parameter determines how much of the selected solid boundaries 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 directions) is connected. Using **Selected boundaries** connects the entire selected boundaries to the shell. If you select **Distance from shell midsurface**, enter a **Distance**  $d$ . This is used instead of the shell thickness for defining the connection distance.

For **Shell boundary** select the name of the corresponding **Solid Connection** node defined at the boundary level in the Shell interface to specify the connected parts.

Select a **Boundary type** — **Shared** or **Parallel**. For **Parallel**, select a **Distance evaluation** — **Shell properties**, **Geometrical distance**, or **User defined**. This parameter determines how the coupling treats the determination of the connection distance. The default is that a distance equal to half the shell thickness is used. If you select **Geometrical distance**, the connection distance is computed from the geometrical distance between the solid boundary and the shell boundary. For **User defined** enter a **Distance**  $d$ . This defines the connection distance.

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Connections>Shell Connection**

*Ribbon*

Physics tab with **Solid Mechanics** selected:



### *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 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.

- If friction is to be included in the modeling, add a [Friction](#) or a [Slip Velocity](#) subnode. The latter provides a simplified modeling when state of the friction and the direction of slip between the two boundaries is known.
- 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 **Slip Velocity** and **Wear** subnodes are only available in the Solid Mechanics and Multibody Dynamics interfaces.

---

Do not select the same contact pair in more than one **Contact** node. Doing so may produce error messages or unpredictable results.

The **Contact** node is only available with some COMSOL products (see <https://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 has 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** check box on the study step **Settings** window is selected and cannot be cleared.

For a Solid Mechanics or Layered Shell interface, a **Contact** node will override all previous nodes in the Model Builder tree sharing the same boundary selections. If you want to add a load (such as the pressure of a surrounding fluid), the best way of doing that is to select a **Boundary Load** from the **Fallback Features** of the **Contact** node. Such a load will then act only on the portions of the boundaries not being in contact. No fallback features are available in the Layered Shell interface.

If an ordinary **Boundary Load** node is added *after* the **Contact** node in the Model Tree, it will contribute to the load on the boundaries, but without taking the contact state into account.



- [Contact Modeling](#)
- [Contact Analysis Theory](#)

## CONTACT SURFACE

If the source side of the contact pair is not part of the current physics interface, select **Source external to current physics**. When selected, only the location of the mesh on the source side is taken into account when evaluating the contact.



If **Source external to current physics** is selected and the source selection intersects with the physics selection, the warning “Source boundary internal to the current physics detected, even though the ‘Source external to current physics’ check box is selected.” is displayed. If you are using a Shell, Layered Shell or Membrane interface on the source boundary, this situation means that any offset or thickness properties given on that selection is ignored by the contact.

The opposite situation, when the source boundary selection is not within the current physics interface, and **Source external to current physics** is not selected, is also problematic. That could cause some variables to become undefined. In this case, the following error is thrown: “Source boundary external to the current physics detected, but the ‘Source external to current physics’ check box is not selected.” If the source selection contains boundaries which are both external and internal to the current physics, an error is always thrown.

When **Contact** is used in a Shell, Layered Shell, and 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.

The setting for the source side is not available if **Source external to current physics** is selected.



The implementation of contact for the Layered Shell interface has some limitations, described under [Contact](#) in [The Layered Shell Interface](#) chapter.



---

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**; or **Augmented Lagrangian, dynamic**. The augmented Lagrangian method is in general more accurate than the penalty method, but also more expensive in terms of computer resources.

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.

### 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.

---



### 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, 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**  $f_{p,init}$  and a **Final penalty factor multiplier**  $f_{p,fin}$ . Also enter a value for the **Transition gap**,  $g_{n,tr}$  to control when to switch between the initial and final penalty factors. The default value is  $\langle phys \rangle . hmin\_dst . * 1e-3$ .
- For **User defined**, enter a **Contact pressure penalty factor**  $p_n$ . The default value is  $\langle phys \rangle . \langle contact\_tag \rangle . E\_char / \langle phys \rangle . hmin\_dst$ . The variable  $E\_char$  contains the value given as characteristic stiffness, and  $hmin\_dst$  is the minimum element size on the destination selection.

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 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**  $\tau_n$ . 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  $(\langle phys \rangle . \langle contact\_tag \rangle . E\_char / \langle phys \rangle . hmin\_dst) * 1 [ms]$ . 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  $(\min(1e-3*(5^{\text{niterCMP}}, 1)) * \langle \text{phys} \rangle . \langle \text{contact\_tag} \rangle . E\_char) / \langle \text{phys} \rangle . hmin\_dst$ . 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*

Select the type of **Penalty factor control** — **Automatic** (default), **Manual tuning**, or **User defined**. These settings are available when **Solution method** is set to **Fully Coupled**.

- 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  $\langle \text{phys} \rangle . \langle \text{contact\_tag} \rangle . E\_char / \langle \text{phys} \rangle . hmin\_dst$ . 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**  $\tau_n$ . 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  $(\langle \text{phys} \rangle . \langle \text{contact\_tag} \rangle . E\_char / \langle \text{phys} \rangle . hmin\_dst) * 1[\text{ms}]$ . 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

This section is only available when **Formulation** is **Augmented Lagrangian** or **Augmented Lagrangian, dynamic**. Select the **Trigger cutback** check box 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 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 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  $\Delta_{\text{gap}}$  are adjusted to be zero before any



offset is added. By default,  $\Delta_{\text{gap}}$  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 box.

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 box.

Typically, the expression 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 expression. By default, the numerical integration of the contact weak equations is two times the shape function order of the displacement field. Deselect the **Use automatic quadrature settings** check box to specify a custom order. When the check box is deselected, 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** both affects the numerical integration of the contact weak contributions and 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 box.

Enter a **Characteristic stiffness**  $E_{\text{char}}$ . 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.

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**  $\Delta_{\text{contact}}$ .



The automatic multiphysics contact detection tolerance is based on the size of the geometric entity.

---

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**

**Shell>Pairs>Contact** (Boundary)

**Layered Shell>Pairs>Contact**

**Membrane>Pairs>Contact**

**Multibody Dynamics>Pairs>Contact**

#### *Ribbon*

Physics tab with **Solid Mechanics**, **Layered Shell**, **Shell**, **Membrane**, or **Multibody Dynamics** selected:

**Pairs>Contact**

#### *Friction*

---

By adding the **Friction** subnode to a **Contact** node, you can model static and dynamic friction.

The selection in the **Friction** node is the same as that of its parent **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

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**  $\mu$ .

For **Exponential dynamic Coulomb** friction, enter a **Static friction coefficient**  $\mu_{\text{stat}}$ , a **Dynamic friction coefficient**  $\mu_{\text{dyn}}$  and a **Friction decay coefficient**  $\alpha_{\text{def}}$ .

The effective friction coefficient is

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}})e^{-\alpha_{\text{def}}\|\mathbf{v}_{\text{slip}}\|}$$

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**  $T_{\text{cohe}}$  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

$$T_{t, \text{crit}} = \min(\mu T_n + T_{\text{cohe}}, T_{t, \text{max}})$$

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  $T_{t, \text{crit}}$ . This expression can depend on any quantity. However, the implementation of the friction model is only correct if  $T_{t, \text{crit}}$  does not depend on the current friction force.

### PENALTY FACTOR

The settings in this section are used to control the penalty factor used by the friction model. The settings available depend on the chosen **Formulation** in the parent **Contact** node.

#### *Settings for the Penalty and the Penalty, Dynamic Methods*

When **Formulation** is **Penalty** or **Penalty, dynamic** in the parent **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  $p_t = p_n/3$ .
- For **User defined**, enter a **Friction force penalty factor**  $p_t$ . The default value is  $(\langle \text{phys} \rangle . \langle \text{contact\_tag} \rangle . E_{\text{char}} / \langle \text{phys} \rangle . h_{\text{min\_dst}}) / 3$ .

#### *Settings for the Augmented Lagrangian Methods*

When **Formulation** is **Augmented Lagrangian** or **Augmented Lagrangian, dynamic** 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** 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** 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** 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** 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**.

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 box.

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 dependent variable and an extra distributed ODE.

You can also request that the total accumulated slip distance is stored by selecting the **Store accumulated slip** check box.



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**

**Shell>Contact>Friction**

**Membrane>Contact>Friction**

**Multibody Dynamics>Contact>Friction**

*Ribbon*

Physics tab with **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 <https://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**  $\mu$ .

For **Exponential dynamic Coulomb** friction, enter a **Static friction coefficient**  $\mu_{\text{stat}}$ , a **Dynamic friction coefficient**  $\mu_{\text{dyn}}$  and a **Friction decay coefficient**  $\alpha_{\text{def}}$ .

The effective friction coefficient is

$$\mu = \mu_{\text{dyn}} + (\mu_{\text{stat}} - \mu_{\text{dyn}})e^{-\alpha_{\text{def}}\|\mathbf{v}_{\text{slip}}\|}$$

where  $\mathbf{v}_{\text{slip}}$  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**  $T_{\text{cohe}}$  to provide an offset to the tangential traction.

Independently, you can enter a **Maximum tangential traction**  $T_{\text{t,max}}$ . 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,  $T_{\text{t,crit}}$ , can be written as

$$T_{\text{t,crit}} = \min(\mu T_{\text{n}} + T_{\text{cohe}}, T_{\text{t,max}})$$


Here,  $T_{\text{n}}$  is the contact pressure.

When **Friction model** is set to **User defined**, enter an expression to define  $T_{\text{t,crit}}$ . This expression can depend on any quantity. However, the implementation of the friction model is only correct if  $T_{\text{t,crit}}$  does not depend implicitly on the current friction force.

## PRESCRIBED VELOCITY

Enter the relative velocity between the destination and source boundaries,  $\mathbf{v}_{\text{slip}}$ . 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 box.

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**  $\Delta_{\text{friction}}$ , 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 is stored by selecting the **Store accumulated slip** check box.



- 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**

**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 contact pair. 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** node. There can only be one **Adhesion** node under a specific **Contact** node. If **Adhesion** and **Friction** are present under the same **Contact** node, the friction settings will be ignored at the locations where the adhesive activation criterion is fulfilled.

## COORDINATE SYSTEM SELECTION

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  $\delta_0$ , 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 when it evaluates to true, adhesion is activated.

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  $n_t$ . For **Adhesive Poisson's ratio**, enter Poisson's ratio for the adhesive layer,  $\nu$ , explicitly.

When **Adhesive stiffness** is set to **User defined**, enter each component of the stiffness vector **k** individually.

When **Adhesive stiffness** is set to **Use material data**, the stiffness vector **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.



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**

**Shell>Contact>Adhesion**

**Membrane>Contact>Adhesion**

**Multibody Dynamics>Contact>Adhesion**

### *Ribbon*

Physics tab with **Contact** selected in the Model Builder tree:

**Attributes>Adhesion**

### *Decohesion*

---

Using the **Decohesion** subnode, you can add decohesion properties to a contact pair. This functionality requires that an **Adhesion** subnode is present and active in the same parent **Contact** 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.

## COORDINATE SYSTEM SELECTION

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**,  $\sigma_t$ . This is the peak stress in pure tension.
- **Shear strength**,  $\sigma_s$ . 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**,  $\lambda$ .


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**  $\alpha$ . 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**,  $\alpha_0$ . The value determines how normal and shear components are combined into a single scalar criterion for damage initiation.
- **Mode mixity exponent**,  $\alpha_c$ . The value determines how normal and shear components are combined into a single scalar failure criterion.
- **Smoothering parameter**,  $N$ . This parameter adjusts the shape of the 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**,  $\tau$ .

#### ADVANCED

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

**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**

**Shell>Contact>Decohesion**

**Membrane>Contact>Decohesion**

**Multibody Dynamics>Contact>Decohesion**

*Ribbon*

Physics tab with **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 a in time-dependent study. For other study types, the **Wear** node is ignored.

The **Wear** node is only available with some COMSOL products (see <https://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  $h_{\text{wear}}$  is added as an offset to the contact condition.

Select a **Wear model** — **Generalized Archard** or **User defined**.

For **Generalized Archard**, enter a **Wear constant**,  $k_{\text{wear}}$ , a **Reference contact pressure**,  $T_{n,\text{ref}}$ , and an **Exponent**,  $n$ . The wear rate is defined as

$$\frac{\partial h_{\text{wear}}}{\partial t} = k_{\text{wear}} \left( \frac{T_n}{T_{n,\text{ref}}} \right)^n \|\mathbf{v}_{\text{slip}}\|$$

Here,  $T_n$  is the contact pressure, and  $\mathbf{v}_{\text{slip}}$  is the slip velocity. For a classical Archard equation,  $n = 1$ . The reference contact pressure,  $T_{n,\text{ref}}$ , can be chosen arbitrarily, and is used only to obtain consistent units. In most cases,  $T_{n,\text{ref}}$  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 classic Archard equation.

Finally, select a **Wear surface** — **Destination** or **Source**. If both surfaces in the contact pair are subject to wear, add 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 Domain](#) material model on the source side.

## ADVANCED

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, except 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.



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**

**Multibody Dynamics>Contact>Wear**

### *Ribbon*

Physics tab with **Contact** selected in the Model Builder tree:

**Attributes>Wear**

## *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.

A default **Bolt Selection** subnode is automatically added. Use one such node to select each individual bolt. Additional **Bolt Selection** subnodes are available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

The **Bolt Pretension** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

### **BOLT PRETENSION**

Select a **Pretension type** — **Pretension force** or **Pretension stress**.

- For **Pretension force** enter a value or expression for  $F_p$ , the pretension force in the bolt.
- For **Pretension stress** enter a value or expression for  $\sigma_p$ , the pretension stress in the bolt. The pretension force is computed through multiplication by the actual area of each selected bolt.



See also

- [Pretensioned Bolts](#) in the Structural Mechanics Modeling chapter for a complete description of how to model pretensioned bolts.
  - [Bolt Pretension Study](#) for information about the **Bolt Pretension** study type.
-

## SOLVER SUGGESTION

---



This section is present only in models created in versions prior to 5.3, in which the new **Bolt Pretension** study type was introduced. The purpose is to maintain backward compatibility for older models.

Select **Solve in bolt pretension study only** check box to make the bolt pretension degrees of freedom behave as in a version 5.3 model, that is being solved for only in a study step of the **Bolt Pretension** type. By default, the check box is cleared, and then the bolt pretension degrees of freedom are solved for in any type of study step, unless you explicitly suppress that.

The settings of the **Solve in bolt pretension study only** check box only affects new study sequences being generated. Existing study sequences will keep the current state for the bolt pretension degrees of freedom.

Once you select the **Solve in bolt pretension study only** check box, the **Solver Suggestion** section will be hidden. Thus, it is not possible to clear the check box again. The degrees of freedom created under this **Bolt Pretension** node from now on assumes the version 5.3 behavior.

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Solid Mechanics>Bolt Pretension**

### *Ribbon*

Physics tab with **Solid Mechanics** 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 bolts. One **Bolt Selection** node is required for each bolt.

## BOUNDARY SELECTION

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.

### **BOLT SELECTION**

Enter a **Bolt label**. The default is `Bolt_1`. The label is used for identification during postprocessing. If the same label is used in a **Bolt Selection** node and a [Thread Boundary Selection](#) node, the results will be grouped together.

If you want to include a **Relaxation** of the bolt pretension, enter a value or expression for  $r$ . The relaxation is an axial displacement, which is subtracted from the predeformation of the bolt.

### **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 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.

---



As default, **Automatic symmetry detection** is selected. When opening models created in versions prior to 5.3, the check box is cleared in order to be compatible with the previous behavior.

---



See also [Pretensioned Bolts](#) in the Structural Mechanics Modeling chapter.

---

### **LOCATION IN USER INTERFACE**

*Context Menus*

**Solid Mechanics>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.

For each bolt, you have to add one **Thread Boundary Selection** subnode to select the contact pair describing each bolt and its matching thread. You can add such subnodes either from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

The **Bolt Thread Contact** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

### **BOLT GEOMETRY**

Enter the **Half thread angle**,  $\alpha$ . 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** check box. Then, enter a vector giving an

**Approximate orientation**,  $\mathbf{e}_{a,\text{approx}}$ . The positive orientation of the bolt axis will be chosen so that  $\mathbf{e}_a \cdot \mathbf{e}_{a,\text{approx}} > 0$ .

- 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,  $(2 * \text{solid}.\text{btc1}.\text{Eequ}) / \text{solid}.\text{btc1}.\text{rd}$ , is intended to supply a stiffness which gives a small elastic slip. This slip is related to the estimated bending stiffness of the threads.

Enter a value for the **Friction coefficient**,  $\mu$ .



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>Bolt Thread Contact**

*Ribbon*

Physics tab with **Solid Mechanics** selected in the Model Builder tree:

**Global>Bolt Thread Contact**

*Thread Boundary Selection*

---

The **Thread Boundary Selection** subnode is used in conjunction with the [Bolt Thread Contact](#) node for selecting the contact pairs representing the internal and external thread. One **Thread Boundary Selection** node is required for each bolt.

## PAIR SELECTION

From the **Pair selection** list, choose the contact pair which 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 LABEL

Enter a **Bolt label**. The default is `Bolt_1`. The label is used for identification during postprocessing. If the same label is used in a **Thread Boundary Selection** node and a **Bolt Selection** node, the results will be grouped together.



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** check box 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.

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Bolt Thread Contact>Thread Boundary Selection**

*Ribbon*

Physics tab with **Bolt Thread Contact** selected in the Model Builder tree:

**Attributes>Thread Boundary Selection**

*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 <https://www.comsol.com/products/specifications/>).

### 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 I

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 internal 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 external 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**

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Boundaries>More>Crack**

*Face Load*

---

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.

## 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.

## 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.

## 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 must 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**,  $r_{\Gamma}$ . 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 external 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.

### 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,  $N_{\Gamma}$ .

When **Relative to path length** is selected, enter a relative number of points along the integration path,  $n_{\Gamma}$ . The total number of integration points is computed as

$$N_{\Gamma} = \frac{r_{\Gamma}}{L_{\text{crack}}} n_{\Gamma}$$

where  $r_{\Gamma}$  is the radius of the integration path, and  $L_{\text{crack}}$  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  $r_\Gamma$  is the radius of the integration path, and  $L_{\text{crack}}$  is the crack length (as represented by the variable `solid.<crack_tag>.crackSize`).



See also [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**

*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/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 <https://www.comsol.com/products/specifications/>).



The **Port** boundary condition exist 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 displacement field  $\mathbf{u}$  (sum of incident and outgoing waves) as

$$\mathbf{u} = \sum_{i \in \text{bnd}} A^{\text{in}} e^{i\phi} (S_{ij} \mathbf{u}_i + \delta_{ij} (\mathbf{u}_i - 2\mathbf{n}(\mathbf{n} \cdot \mathbf{u}_i)))$$

where the summation “i” is over all ports on the given boundary “bnd”,  $S_{ij}$  is the scattering parameter,  $A^{\text{in}}$  is the amplitude of the incident field (at port “j”),  $\phi$  is a possible phase, and  $\mathbf{u}_i$  is the mode shape of the i-th port. The mode shape  $\mathbf{u}_i$  is normalized to carry a unit power of 1 W. This definition means that the scattering parameter  $S_{ij}$  defines the amplitude of mode i when a system is excited at port j (with mode j). This corresponds to a multi-mode 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`, etc. 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.

## 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  $u_n$**  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  $A^{\text{in}}$  of the incident wave. This is in general defined as the maximum amplitude for a given mode shape.
- For **Power** enter the power  $P^{\text{in}}$  of the incident wave. This is in general defined as the RMS power of the incident wave.
- For both options enter the phase  $\phi$  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 etc.

---

## 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` etc.



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**

*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 in 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).

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 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 check box **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 check box **Setup deformed geometry** is checked (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)` etc. 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 check box 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 larger 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.

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>More>Elastic Predeformation** (Domain)

*Ribbon*

Physics tab with **Solid Mechanics** selected:

**Domains>More>Elastic Predeformation**

## 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 ( $\phi$ ), 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,  $f$ :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

### 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**  $\phi$  in radians (for a pressure the load phase  $\phi$  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 Domain>Applied Force>Phase**  
**Solid Mechanics>Rigid Domain>Applied Moment>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 Domain>Applied Force>Phase**  
**Layered Shell>Rigid Domain>Applied Moment>Phase**  
**Membrane>Body Load>Phase**  
**Membrane>Face Load>Phase**  
**Membrane>Edge 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 Domain>Applied Force>Phase**  
**Multibody Dynamics>Rigid Domain>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 Domain>Prescribed Displacement/Rotation>Harmonic Perturbation**  
**Solid Mechanics>Rigid Connector>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**  
**Multibody Dynamics>Prescribed Displacement>Harmonic Perturbation**

**Multibody Dynamics>Rigid Domain>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.

When a **Stress Linearization** node has been added, 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 line selection can consist of several edges, but they are assumed to form a straight line from one free boundary to another. For 2D and 2D Axisymmetric geometries, the only input is the line selection. For 3D, you also need to specify the orientation of the local coordinate system in which the linearized stresses are represented.

The **Stress Linearization** node is only available with some COMSOL products (see <https://www.comsol.com/products/specifications/>).

### **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 line and the second direction.

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.



See also

- [Stress Linearization](#) in the Structural Mechanics Modeling Chapter.
  - [Stress Linearization](#) in the Structural Mechanics Theory Chapter.
- 

## LOCATION IN USER INTERFACE

*Context Menus*

**Solid Mechanics>Variables>Stress Linearization**

*Ribbon*

Physics tab with **Solid Mechanics** selected in the Model Builder tree:

**Global>Variables>Stress Linearization**

*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. Add 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** or **Finite rotations**. With **Small rotations**, the computation of displacements caused by rotations is linearized using a cross product. If large rotations are expected, the **Finite rotation** formulation is more accurate. In this case, the rotation is described by a nonlinear rotation matrix.

## FORMULATION

Select a **Formulation** — **Symmetric segregated** or **Unsymmetric coupled**. This affects how the least-squares equations for computing the average rotations are formulated. Using an appropriate setting here, and a corresponding setup of the solver strategy can strongly affect the computational cost. The cost of solving for the average rotation variables as such is very small, but it can affect the structure of the system of equations.

With the **Unsymmetric coupled** option, you do not have to care about solver setup. However, the majority of structural mechanics problems produce a symmetric stiffness matrix, and that symmetry is now broken by adding the equations for 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 rotations and displacements are 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 rotations. 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 and displacement 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 and displacement 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**

**Multibody Dynamics>Variables>Average Rotation**

**Solid Rotor>Variables>Average Rotation**

### *Ribbon*

Physics tab with **Solid Mechanics** selected in the Model Builder tree:

**Global>Variables>Average Rotation**



## 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 <https://www.comsol.com/products/specifications/>).

### 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 **Coordinate system axes** is 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.

---

### LOCATION IN USER INTERFACE

#### Context Menus

**Solid Mechanics>Linear Elastic Material>Variables>Wave Speeds**

**Solid Mechanics>Piezoelectric 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 4-13](#), 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 4-13: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.disp	Total displacement	All materials <sup>1</sup>	
solid.vel	Velocity magnitude	All materials <sup>1</sup>	
solid.u_tX	Velocity, X component	All materials <sup>1</sup>	
solid.acc	Acceleration magnitude	All materials <sup>1</sup>	

TABLE 4-13: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.u_ttX	Acceleration, X component	All materials <sup>1</sup>	
solid.eXX	Strain tensor, XX component	All materials <sup>1</sup>	Total (Green–Lagrange) strain in global directions. See also <a href="#">Deformation Measures</a> .
solid.e111	Strain tensor, local coordinate system, 11 component	All materials <sup>1</sup>	Total strain in the local directions.
solid.eel11	Elastic strain tensor, local coordinate system, 11 component	All materials <sup>1</sup>	
solid.eeldev11	Deviatoric elastic strain tensor, local coordinate system, 11 component	All materials <sup>1</sup>	
solid.eiel11	Inelastic strain tensor, local coordinate system, 11 component	All materials <sup>1</sup>	See also <a href="#">Inelastic Strain Contributions</a> .
solid.FdxX	Deformation gradient, xX component	All materials <sup>1</sup>	See also <a href="#">Deformation Measures</a> .
solid.Fdlx1	Deformation gradient, local, x1 component	All materials <sup>1</sup>	
solid.J	Volume ratio	All materials <sup>1</sup>	Total relative volume change.
solid.Cel11	Elastic Cauchy-Green tensor, local coordinate system, 11 component	All materials <sup>1</sup>	
solid.Jel	Elastic volume ratio	All materials <sup>1</sup>	Elastic relative volume change.
solid.Ldxy	Rate of strain tensor, xy component	All materials <sup>1</sup>	See also <a href="#">Strain Rate and Spin</a> .

TABLE 4-13: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.Lwxy	Spin tensor, xy component	All materials <sup>1</sup>	See also <a href="#">Strain Rate and Spin</a> .
solid.SXX	Second Piola–Kirchhoff stress, XX component	All materials <sup>1</sup>	See also <a href="#">Defining Stress</a> .
solid.S111	Second Piola–Kirchhoff stress, local coordinate system, II component	All materials <sup>1</sup>	
solid.SdevX	Deviatoric second Piola–Kirchhoff stress, X component	All material <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.PxX	First Piola–Kirchhoff stress, xX component	All materials <sup>1</sup>	See also <a href="#">Defining Stress</a> .
solid.sxy	Stress tensor, xy component	All materials <sup>1</sup>	Cauchy stress, See also <a href="#">Defining Stress</a> .
solid.sdevx	Deviatoric stress tensor, x component	All materials <sup>1</sup>	Deviatoric Cauchy stress, See also <a href="#">Invariants of the Stress Tensor</a> .
solid.mises	von Mises stress	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.tresca	Tresca stress	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.I1s	First principal invariant of stress	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.I2s	Second principal invariant of stress	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.I3s	Third principal invariant of stress	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .


TABLE 4-13: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.II2s	Second invariant of stress deviator	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.II3s	Third invariant of stress deviator	All materials <sup>1</sup>	See also <a href="#">Invariants of the Stress Tensor</a> .
solid.RFx	Reaction force, x component	All materials <sup>1</sup>	
solid.RMx	Reaction moment, y component	All materials <sup>1</sup>	
solid.Tax	Traction (force/area), x component	All materials <sup>1</sup>	

<sup>1</sup> - All materials: **Linear Elastic Material, Nonlinear Elastic Material, Hyperelastic Material, Piezoelectric Material, Magnetostrictive Material, Cam-Clay Material.**



## Shell and Plate

This chapter describes the Shell and Plate interfaces, which are found under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Modeling with Plates and Shells](#)
- [Theory for the Shell and Plate Interfaces](#)
- [The Shell and Plate Interfaces](#)

# Modeling with Plates and Shells

In this section:

- [Introduction](#)
- [Dependent Variables](#)
- [Layered and Nonlayered Shells](#)
- [Working with Section Stiffness](#)

## *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. 1](#).

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.

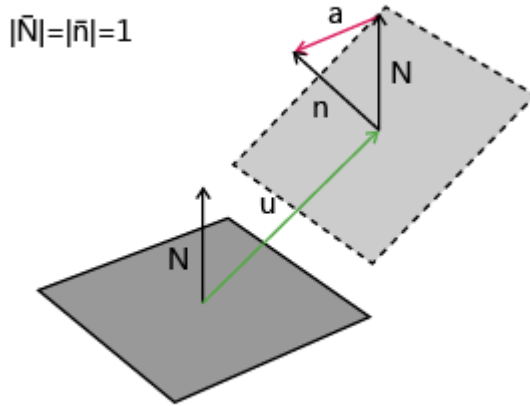


Figure 5-1: The degrees of freedom in the shell interface.  $\mathbf{N}$  is the normal vector in the original configuration and  $\mathbf{n}$  is the normal in the deformed state.

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,  $\mathbf{u}$ , 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  $\zeta_0$  is the offset. The rotational displacement  $\mathbf{a}$  is the same on both midsurface and reference surface.



In axisymmetry, there are four degrees of freedom, since  $\mathbf{u}$  and  $\mathbf{a}$  only has 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  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$  about the global axes. For a geometrically linear analysis, the relation between normal displacement and rotation vector is simple:  $\mathbf{a} = \boldsymbol{\theta} \times \mathbf{n}$  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  $a_x$  and  $a_y$ . 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  $\theta_x$ ,  $\theta_y$ , and (possibly)  $\theta_z$  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 Domain](#), and [Section Stiffness](#). The other type consists of [Layered Linear Elastic Material](#) and [Layered Hyperelastic Material](#). 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 **Layered Linear Elastic Material** and **Layered Hyperelastic Material** 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, the **Layered Linear Elastic Material** forms the basis for all nonlinear material models even if the shell is not

layered as such. Also, if you want to write your own expressions as function of through-thickness location, you must use this material model.

When the Nonlinear Structural Materials Module is available, the **Layered Linear Elastic Material** 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.



The **Layered Linear Elastic Material** is only available for the Shell interface, not for the Plate interface.

The accuracy of the results 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 *extra 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 5-1](#), the number of integration points that are used in the thickness direction are summarized.

TABLE 5-1: 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, etc.



---

Actually, the **Layered Linear Elastic Material** 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.

---

In order to use the section stiffness approach, you must obtain all terms in the stiffness, and in the case of dynamic analysis, inertial matrices. 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  $\epsilon$  to compute stresses. The stress state will depend on the local geometry. In the **Section Stiffness** node, however, you have a possibility to define expressions for the relation between section forces and stresses. Such expressions will then be based on your knowledge of the peak stresses in the nonhomogenized structure.

### *Loads*

---

Loads are assumed to be applied at the midsurface. This has some important implications:

- If the shell is curved and thick, the total force produced by a load on the midsurface can differ significantly from what is obtained when the true load is acting on the inner or outer surface. You may thus need to scale the physical load, depending on where it is applied. For example, if you have a circular pipe with inner and outer diameters  $D_i$  and  $D_o$  respectively, an internal pressure  $p$  should be scaled as  $2pD_i/(D_i + D_o)$  to give a consistent midsurface load.
- If a load  $F_t$  acts tangentially on the top or bottom surface, then you should add also a moment load with the magnitude  $F_t d/2$ , since the distance to the midsurface is half the thickness  $d$ . The sign of the moment contribution depends on whether the

load is applied to the top or the bottom side of the shell. This effect is present also for flat shells.

# 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 Shell Interface](#)

## *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](#)

### **GEOMETRY AND DEFORMATION**

Let  $\mathbf{r}$  be the undeformed shell midsurface position,  $\xi^i$  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  $\xi^1$  and  $\xi^2$  follow the midsurface, and  $\xi^3$  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

$$|\mathbf{n} + \mathbf{a}| = 1 \quad (5-1)$$

In a geometrically linear analysis Equation 5-1 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  $n$ th-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:  $\xi^1$  is the arc-length along the line on which the shell is modeled, and  $\xi^2$  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  $\alpha$  and  $\beta$  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  $\varepsilon_{33}$  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  $\omega_\alpha$  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  $\zeta_\alpha$  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,

$$\mathbf{r}(\xi^1, \xi^2) = \mathbf{r}_R(\xi^1, \xi^2) + \zeta_0 \mathbf{n}(\xi^1, \xi^2)$$

where  $\mathbf{r}_R$  is the position of the meshed *reference surface* and  $\zeta_0$  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:

$$\frac{\partial \mathbf{r}}{\partial \xi^\alpha} = \frac{\partial \mathbf{r}_R}{\partial \xi^\alpha} + \zeta_0 \frac{\partial \mathbf{n}}{\partial \xi^\alpha}$$

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, 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 is required. It is 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.

## ROTATION REPRESENTATION

In a geometrically linear analysis, a rotation vector is defined as

$$\boldsymbol{\theta} = \mathbf{n} \times \mathbf{a}$$

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

$$|\theta| = \arccos(1 + \mathbf{n} \cdot \mathbf{a})$$

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. 1) 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. This is automatically handled by the program. 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 3 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

$$\mathbf{n}_j \cdot \mathbf{a}_k + \mathbf{a}_j \cdot \mathbf{n}_k + \mathbf{a}_j \cdot \mathbf{a}_k = 0 \quad (5-2)$$

where the values of  $j$  and  $k$  range over the number of shells elements with different normals. The third term in Equation 5-2 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

$$\int_V f(\xi^1, \xi^2, \xi^3) dV = \int_{A-d/2}^{d/2} \int_A f_1(\xi^1, \xi^2) f_2(\xi^3) d\xi dA = \int_A f_1(\xi^1, \xi^2) F_2(d) dA$$

All functions of  $\zeta$  are assumed to be of the form  $\zeta^n$ . 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 which 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

$$\mathbf{a}_0 = \boldsymbol{\Omega}_0 \times \mathbf{n}$$

where  $\boldsymbol{\Omega}_0$  is the vector of prescribed nodal rotations. This relation is fully defined only when all three components of  $\boldsymbol{\Omega}_0$  are given. It is also possible to prescribe only one or two of the components of  $\boldsymbol{\Omega}_0$ , 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 to 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  $\Omega_{0i}$  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

$$\mathbf{a}_0 = (\mathbf{R}(\boldsymbol{\Omega}_0) - \mathbf{I})\mathbf{n}$$

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  $\boldsymbol{\omega}$  as

$$\dot{\mathbf{a}} = \boldsymbol{\omega} \times (\mathbf{n} + \mathbf{a}_0)$$

### **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 ( $\gamma_{\alpha\beta}$ ,  $\chi_{\alpha\beta}$ ,  $\zeta_{\alpha}$ ) 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

$$\sigma_m = D[\gamma - \gamma_i - \alpha(T_m - T_{ref}) - \beta_h(c_m - c_{mo,ref})] + \frac{N_i}{d}$$

where  $D$  is the plane stress constitutive matrix,  $N_i$  are the initial membrane forces, and  $\gamma_i$  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

$$\sigma_b = \frac{Dd}{2} \left[ \chi - \chi_i - \alpha \frac{\Delta T}{d} - \beta_h \frac{\Delta c_{mo}}{d} \right] + \frac{6M_i}{d^2}$$

where  $\chi_i$  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.  $\Delta T$  is the temperature difference between the top and bottom surface of the shell, and  $\Delta c_{mo}$  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  $G$  represent the transverse shear moduli,  $\zeta_i$  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  $\zeta$  and  $\zeta_i$  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

$$N = d\sigma_m$$

$$M = \frac{d^2}{6}\sigma_b$$

$$Q = d\sigma_s$$

## 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 is also possible to control the orientation of the shell normal by selecting the **Reverse normal direction** check box.

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

$$\mathbf{e}_{x1} = \frac{\mathbf{e}_{x1} - (\mathbf{e}_{x1} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}}{|\mathbf{e}_{x1} - (\mathbf{e}_{x1} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}|}$$

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

$$\mathbf{e}_{x1} = \frac{\mathbf{e}_{x2} - (\mathbf{e}_{x2} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}}{|\mathbf{e}_{x2} - (\mathbf{e}_{x2} \cdot \mathbf{e}_{z1})\mathbf{e}_{z1}|}$$

3. At last, 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  $\mathbf{x}_1$ ,  $\mathbf{y}_1$ , and  $\mathbf{z}_1$  are defined so that:

- The first direction ( $\mathbf{x}_1$ ) is along the edge. This direction can be visualized by selecting the **Show edge directions arrows** check box in the **View** settings.
- The third direction ( $\mathbf{z}_1$ ) is the same as the shell normal direction. The shell normal direction can be visualized in the default plot named *Undeformed geometry*, or by adding a **Coordinate System Surface** plot and selecting the **Shell, Local System**.
- The second direction ( $\mathbf{y}_1$ ) is in the plane of the shell and orthogonal to the edge. It is formed by the cross product of  $\mathbf{z}_1$  and  $\mathbf{x}_1$ ;  $\mathbf{y}_1 = \mathbf{z}_1 \times \mathbf{x}_1$ .

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  $\mathbf{z}_1$  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  $z_{\text{shell}}$  (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 default plots, 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 ( $I$ ) of a rigid shell take the finite thickness into account. They are computed as:

$$\begin{aligned}
m &= \int_V \rho dV = \int_A \rho d A \\
\mathbf{X}_M &= \frac{1}{m} \int_V \rho \mathbf{X} dV = \frac{1}{m} \int_V \rho (\mathbf{r} + \zeta \mathbf{n}) dV = \frac{1}{m} \int_A \rho d\mathbf{r} dA \\
I &= \int_V (((\mathbf{X} - \mathbf{X}_M)^T \cdot (\mathbf{X} - \mathbf{X}_M)) E_3 - (\mathbf{X} - \mathbf{X}_M) \cdot (\mathbf{X} - \mathbf{X}_M)^T) \rho dV = \\
&\int_A (((\mathbf{r} - \mathbf{X}_M)^T \cdot (\mathbf{r} - \mathbf{X}_M)) E_3 - (\mathbf{r} - \mathbf{X}_M) \cdot (\mathbf{r} - \mathbf{X}_M)^T) \rho d A + \\
&\int_A (E_3 - \mathbf{n} \mathbf{n}^T) \rho \frac{d^3}{12} dA
\end{aligned}$$

where  $E_3$  and  $\mathbf{X}_M$  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

---

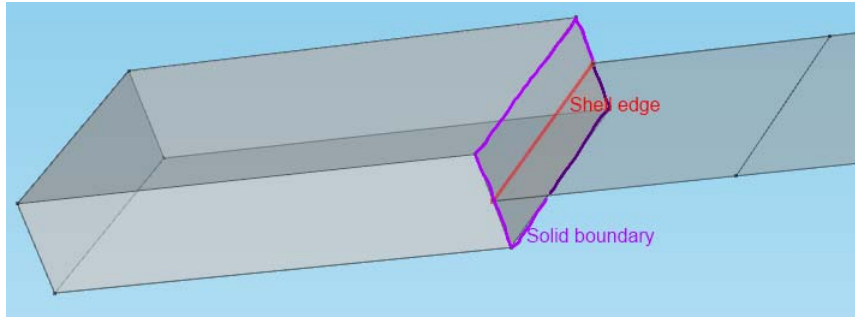
There are three types of connections between a shell and a solid of interest:

- *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 5-2 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 5-2: An example of a shell extending perpendicular to the solid boundary.*

The connection of a solid to a shell is based on that shell theory is valid on both sides of the connection. This can be divided into these 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:

$$\begin{aligned} u'_{sld}(y',z') &= u'_{sh}(y',0) + z'\alpha'_x(y',0) \\ v'_{sld}(y',z') &= v'_{sh}(y',0) + z'\alpha'_y(y',0) \end{aligned}$$

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

$$w'_{sld}(y',z') = w'_{sh}(y',0)$$

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

$$\left\{ \begin{array}{l} \sigma_{x'} = A_1 + B_1 z' \\ \sigma_{y'} = A_2 + B_2 z' \\ \sigma_{z'} = 0 \\ \tau_{x'y'} = B_3 z' \\ \tau_{x'z'} = C_1(1 - z'^2) \\ \tau_{y'z'} = C_2(1 - z'^2) \end{array} \right.$$

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

$$\varepsilon_{z'} = -\frac{\nu}{E}(\sigma_{x'} + \sigma_{y'}) = K_1(x',y') + 2K_2(x',y')z'$$

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

$$\begin{aligned} u'_{\text{slid}}(y',z') &= u'_{\text{sh}}(y',0) + z'a'_{x'}(y',0) + q_{\text{u}}(y',0)(z^3 - k_{\text{u}}z)d \\ v'_{\text{slid}}(y',z') &= v'_{\text{sh}}(y',0) + z'a'_{y'}(y',0) \\ w'_{\text{slid}}(y',z') &= w'_{\text{sh}}(y',0) + q_{\text{w1}}(y',0)zd + q_{\text{w2}}(y',0)(z^2 - k_{\text{w2}})d \end{aligned}$$

Here  $q_{\text{u}}$ ,  $q_{\text{w1}}$ , and  $q_{\text{w2}}$  are the new dependent variables defined on the shell edge. They are dimensionless, due to the multiplication with the shell thickness,  $d$ . The constants  $k_{\text{u}} = 3/5$  and  $k_{\text{w2}} = 1/5$  are explained below. The variable  $q_{\text{w1}}$  is proportional to the membrane axial strain, the variable  $q_{\text{w2}}$  is proportional to the bending strain, and the variable  $q_{\text{u}}$  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  $q_{\text{u}}$ ,  $q_{\text{w1}}$ , and  $q_{\text{w2}}$  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:

$$\begin{aligned} &\frac{d}{2} \\ \text{test}(q_{\text{u}}) \int &(z^3 - k_{\text{u}}z)\sigma_{x'}dz' = 0 \\ &\frac{-d}{2} \\ &\frac{d}{2} \\ \text{test}(q_{\text{w1}}) \int &z\tau_{x'z'}dz' = 0 \\ &\frac{-d}{2} \\ &\frac{d}{2} \\ \text{test}(q_{\text{w2}}) \int &(z^2 - k_{\text{w2}})\tau_{x'z'}dz' = 0 \\ &\frac{-d}{2} \end{aligned}$$

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  $k_u$  and  $k_{w2}$ .

The constraint expressions must now be formulated in global directions. As a start, the constraints are written on vector form in local directions as

$$\mathbf{u}'_{sld} = \mathbf{u}'_{sh} + z'\mathbf{a}' + \mathbf{q} \quad (5-3)$$

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 5-3](#).

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. Under geometric nonlinearity, the computation should be based on the current geometry.

$$z' = (\mathbf{x}_{sld} - \mathbf{x}_{sh}) \cdot \mathbf{n}_{sh}$$

The latter expression introduces additional nonlinearities in the model because it depends on deformed position and deformed normal. Also, the position of the shell midsurface with respect to the solid is actually part of the solution.

Let  $\Phi$  be the matrix that transforms displacements from the global system to the local system:

$$\Phi = [\mathbf{e}_{x'} \ \mathbf{e}_{y'} \ \mathbf{e}_{z'}]$$

The expression for the constraints in global directions then becomes

$$\mathbf{u}_{\text{sld}} = \Phi^T \mathbf{u}'_{\text{sld}} = \Phi^T (\mathbf{u}'_{\text{sh}} + z' \mathbf{a}' + \mathbf{q}) = \mathbf{u}_{\text{sh}} + z' \mathbf{a} + \Phi^T \mathbf{q}$$

The only transformation actually needed is thus the projection of the  $\mathbf{q}$  vector. For a linear case, the transformation can be written as

$$\Phi = \begin{bmatrix} s \mathbf{t}_1 \times \mathbf{N}_{\text{sh}} & s \mathbf{t}_1 & \mathbf{N}_{\text{sh}} \end{bmatrix}$$

where  $\mathbf{N}$  is the undeformed shell normal (`shell.an`) and  $\mathbf{t}_1$  is the shell edge tangent (`shell.t1e`). The coefficient  $s$  is either 1 or  $-1$ , and 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 is 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

$$\mathbf{u}_{\text{sld}} = \mathbf{u}_{\text{shl}} - \zeta \mathbf{a}$$

where  $\zeta$  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  $\zeta$  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

$$(\theta_b - \mathbf{n} \times \mathbf{a}) \cdot \mathbf{t}_i \quad i = 1, 2$$

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

$$\begin{aligned}\theta_b \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 &= 0 \\ \theta_b \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 &= 0\end{aligned}$$

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.

### *Theory for FSDT Laminated Shell*

---

The theory of laminated shells is discussed in this section. The **Layered Linear Elastic Material** node in the Shell interface allows the modeling of laminated shells, also popularly 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 a composite laminate.

Several topics are discussed in this section:

- [About Composite Laminates](#)
- [Equivalent Single Layer \(ESL\) Theory](#)
- [First-Order Shear Deformation Theory \(FSDT\)](#)
- [Integration in a Laminate](#)

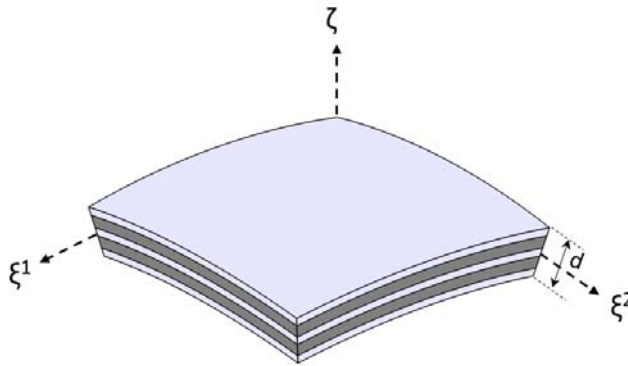
- Lamina Constitutive Law
- Laminate Constitutive Law
- Shear Correction Factor Computation
- Results Evaluation in a Laminate

#### 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.

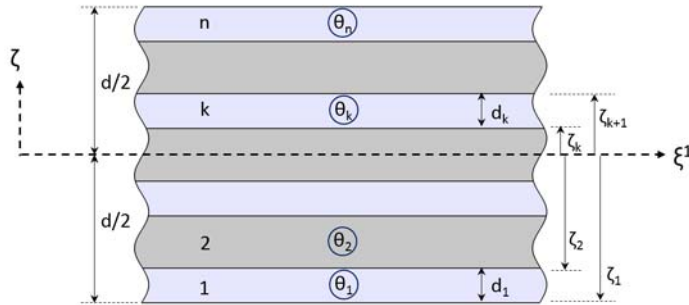


*Figure 5-3: Geometry of a doubly curved composite laminate.*

Figure 5-3 shows a doubly curved laminated shell with uniform total thickness,  $d$ . It is represented by orthogonal curvilinear coordinate system  $(\xi^1, \xi^2, \zeta)$ . 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 5-4](#). The thickness of each layer ( $d_k$ ), as well as the fiber direction in each layer ( $\theta_k$ ) with respect to the first principal direction ( $\xi^1$ ) of the laminate are indicated. A counterclockwise rotation of the fiber direction with respect to the  $\zeta$  direction is considered as positive.



*Figure 5-4: 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:

$$\int_V f(\xi^1, \xi^2, \zeta) dV = \int_A \int_{\zeta} f_1(\xi^1, \xi^2) f_2(\zeta) d\zeta dA = \int \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} f_1(\xi^1, \xi^2) f_2(\zeta) dA$$

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 ( $\sigma_{xz}$ ,  $\sigma_{yz}$ ,  $\sigma_{zz}$ ) are assumed to be zero. In terms of strains, the transverse shear strain components ( $\epsilon_{xz}$ ,  $\epsilon_{yz}$ ) are zero while the transverse normal strain ( $\epsilon_{zz}$ ) 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:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}^{(k)} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{bmatrix}^{(k)}$$

where

- $\sigma_{ij}$  are the stress components in the principal material directions of a lamina
- $\epsilon_{ij}$  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 ( $\xi^1, \xi^2, \zeta$ ) as shown in [Figure 5-3](#). The constitutive equations for a lamina can be transformed from its material coordinate system to the laminate coordinate system, giving:

$$\begin{bmatrix} \sigma_{\xi^1\xi^1} \\ \sigma_{\xi^2\xi^2} \\ \sigma_{\xi^1\xi^2} \end{bmatrix}^{(k)} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^1\xi^1} \\ \varepsilon_{\xi^2\xi^2} \\ \varepsilon_{\xi^1\xi^2} \end{bmatrix}^{(k)}$$

where

- $\sigma_{ij}$  are the stress components in the laminate coordinate system
- $\varepsilon_{ij}$  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:

$$\bar{Q} = T^{-1}QT$$

The principal material direction (or fiber direction) in each lamina makes an angle ( $\theta$ ) with the first in-plane direction ( $\xi^1$ ) 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

- $\sigma_{ij}$  are the transverse shear stress components in the principal material directions of a lamina
- $\varepsilon_{ij}$  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:

$$\begin{bmatrix} \sigma_{\xi^2\xi^3} \\ \sigma_{\xi^1\xi^3} \end{bmatrix}^{(k)} = \begin{bmatrix} \bar{Q}_{44} & \bar{Q}_{45} \\ \bar{Q}_{45} & \bar{Q}_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^2\xi^3} \\ \varepsilon_{\xi^1\xi^3} \end{bmatrix}^{(k)}$$

where

- $\sigma_{ij}$  are the transverse shear stress components in the laminate coordinate system
- $\varepsilon_{ij}$  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:

$$\bar{Q} = T^{-1}QT$$

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.

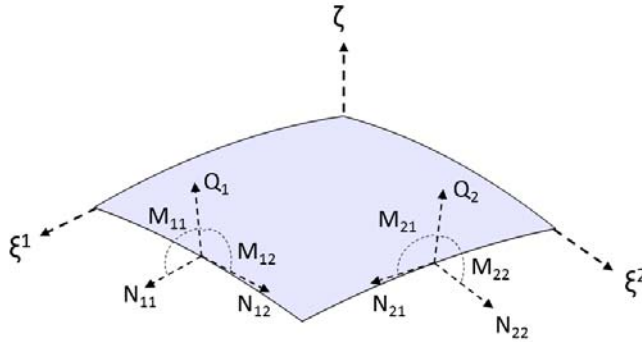


Figure 5-5: An equivalent layer of a composite laminate having  $n$  layers. The resultant membrane forces ( $N_{ij}$ ), bending moments ( $M_{ij}$ ), and transverse shear forces ( $Q_i$ ) are shown.

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
- $\sigma_{ij}$  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:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1 \xi^1}^{(k)} \\ \sigma_{\xi^2 \xi^2}^{(k)} \\ \sigma_{\xi^1 \xi^2}^{(k)} \end{bmatrix} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1 \xi^1}^{(k)} \\ \sigma_{\xi^2 \xi^2}^{(k)} \\ \sigma_{\xi^1 \xi^2}^{(k)} \end{bmatrix} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^2 \xi^3}^{(k)} \\ \sigma_{\xi^1 \xi^3}^{(k)} \end{bmatrix} d\zeta$$

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:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^1 \xi^1}^{(0)} + \zeta \varepsilon_{\xi^1 \xi^1}^{(1)} \\ \varepsilon_{\xi^2 \xi^2}^{(0)} + \zeta \varepsilon_{\xi^2 \xi^2}^{(1)} \\ \varepsilon_{\xi^1 \xi^2}^{(0)} + \zeta \varepsilon_{\xi^1 \xi^2}^{(1)} \end{bmatrix} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^1 \xi^1}^{(0)} + \zeta \varepsilon_{\xi^1 \xi^1}^{(1)} \\ \varepsilon_{\xi^2 \xi^2}^{(0)} + \zeta \varepsilon_{\xi^2 \xi^2}^{(1)} \\ \varepsilon_{\xi^1 \xi^2}^{(0)} + \zeta \varepsilon_{\xi^1 \xi^2}^{(1)} \end{bmatrix} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \bar{Q}_{44} & \bar{Q}_{45} \\ \bar{Q}_{45} & \bar{Q}_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^2 \xi^3}^{(0)} \\ \varepsilon_{\xi^1 \xi^3}^{(0)} \end{bmatrix} d\zeta$$



The resultant force and moment expressions can be further rewritten as:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(0)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(0)} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(1)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(1)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(1)} \end{bmatrix} \quad (5-4)$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(0)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(0)} \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(1)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(1)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(1)} \end{bmatrix} \quad (5-5)$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \begin{bmatrix} A_{44} & A_{45} \\ A_{45} & A_{55} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^2 \zeta^3}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^3}^{(0)} \end{bmatrix} \quad (5-6)$$

where

- $\varepsilon_{ij}^{(0)}$  is the membrane part of strain
- $\varepsilon_{ij}^{(1)}$  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:

$$A_{ij} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \bar{Q}_{ij}^{(k)} d\zeta = \sum_{k=1}^n \bar{Q}_{ij}^{(k)} (\zeta_{k+1} - \zeta_k)$$

$$B_{ij} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \bar{Q}_{ij}^{(k)} \zeta d\zeta = \frac{1}{2} \sum_{k=1}^n \bar{Q}_{ij}^{(k)} (\zeta_{k+1}^2 - \zeta_k^2)$$

$$D_{ij} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \bar{Q}_{ij}^{(k)} \zeta^2 d\zeta = \frac{1}{3} \sum_{k=1}^n \bar{Q}_{ij}^{(k)} (\zeta_{k+1}^3 - \zeta_k^3)$$

The in-plane laminate constitutive law, relating resultant forces and midplane strains, can also be written in matrix form as:

$$\begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{bmatrix} \{\epsilon^{(0)}\} \\ \{\epsilon^{(1)}\} \end{bmatrix}$$

The midplane strains can be written using resultant forces and flexibility matrices as:

$$\begin{bmatrix} \{\epsilon^{(0)}\} \\ \{\epsilon^{(1)}\} \end{bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix}^{-1} \begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix}$$

$$\begin{bmatrix} \{\epsilon^{(0)}\} \\ \{\epsilon^{(1)}\} \end{bmatrix} = \begin{bmatrix} [a] & [b] \\ [b] & [d] \end{bmatrix} \begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix}$$

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,  $\bar{Q}_{45} = \bar{Q}_{54} = 0$ .
- 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:

$$\begin{aligned}\sigma_1^{(\text{bot})} &= 0 \\ \sigma_i^{(\text{top})} &= \sigma_{i+1}^{(\text{bot})} \\ \sigma_n^{(\text{top})} &= 0\end{aligned}$$

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:

$$U_s = \frac{1}{2} \iint_A \int_{\zeta} \sigma \gamma d\zeta dA = \frac{1}{2} \iint_A \int_{\zeta} 4f(\zeta)^2 G(\zeta) \varepsilon_{\text{avg}}^2 d\zeta dA$$

where the shear strain distribution is defined as:

$$f(\zeta) = E\zeta^2 + F\zeta + K_i$$

The transverse shear strain energy, corresponding to  $(\varepsilon_{13})$ , based on FSDT theory is written as:

$$U_s = \frac{1}{2K_{13}} \int_A 2Q_1 \varepsilon_{\text{avg}} dA = \frac{1}{2K_{13}} \int_A 4A_{55} \varepsilon_{\text{avg}}^2 dA$$

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  $(\varepsilon_{23})$  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 different elastic strain energy expressions. Several predefined material models are included, and also the option to enter user defined expressions for the strain energy density. See [Hyperelastic Material Models](#) in the *Structural Mechanics Theory* chapter.

### *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 etc. 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 of 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 5-4](#) to [Equation 5-6](#) 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

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \mathbf{S}_N \begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} + \mathbf{S}_M \begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix}$$

$$\begin{bmatrix} \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \mathbf{S}_Q \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$



where matrices  $\mathbf{S}_N$ ,  $\mathbf{S}_M$ , and  $\mathbf{S}_Q$  are user inputs. They are by default populated with coefficient that gives the stress on the top surface of the shell.



### *References for the Shell Interface*

---

1. D. Chapelle and K.J. Bathe, *The Finite Element Analysis of Shells — Fundamentals*, Springer-Verlag, Berlin Heidelberg, 2003.
2. J.N. Reddy, *Mechanics of Laminated Composite Plates and Shells — Theory and Analysis*, CRC Press, Second Edition, 2004.

# 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 is shown.

### 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.

### REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

### FOLD-LINE LIMIT ANGLE

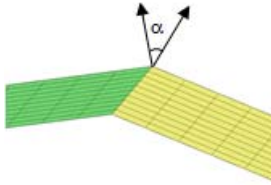


This section is available for the Shell interface only. Also see [The MITC Shell Formulation](#).

---

The fold-line limit angle  $\alpha$  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  $\alpha$  field. The default

value is 0.05 radians (approximately 3 degrees). The value must be larger than 0, and less than  $\pi/2$ , but angles larger than a few degrees are not usually meaningful.




Since the rotational degrees of freedom have different meaning across a fold line, they are separate degrees of freedom, which are joined by a constraint. This constraint is, as default, implemented as a pointwise constraint. Select **Use weak constraints for fold-lines** to use a weak constraint instead.

#### 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 box. Normally these settings do not need to be changed.

The **Use MITC interpolation** check box 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** check box 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 check box is selected by default.

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 **Use weak constraints for shell normals** to use a weak constraint instead.

## 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 nodes. This means that it can, and often does, connect elements from different physics interfaces.

In the drop-down 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
- Antisymmetry
- Attachment
- Beam Connection
- Body Load
- Damping
- Edge Load
- External Stress
- Face Load
- Fixed Constraint
- Hygroscopic Swelling
- Initial Values
- Initial Stress and Strain
- Layered Linear Elastic Material
- Layered Hyperelastic Material
- Linear Elastic Material
- No Rotation
- Periodic Condition
- Phase
- Pinned
- Point Load
- Point Mass
- Point Mass Damping
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Rigid Connector
- Ring Load
- Safety
- Section Stiffness
- Shell Local System
- Simply Supported
- Solid Connection
- Spring Foundation
- Symmetry
- Symmetry Plane
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Attachment)
- Thermal Expansion (Rigid Connector)
- 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)
- Center of Rotation Nodes
- Contact
- Creep
- Damage
- Free
- Friction
- Mass and Moment of Inertia (Rigid Connector)
- Gravity<sup>1</sup>
- Plasticity
- Point Load (on Axis)
- Rigid Domain
- Rotating Frame<sup>1</sup>
- Spring-Damper
- Viscoelasticity
- Viscoplasticity

<sup>1</sup> 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*

---



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 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  $\frac{\partial \mathbf{u}}{\partial t}$**
- **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 the shell 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 shell 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 shell midsurface is displaced from the meshed boundary in the direction of the positive shell normal.

## THICKNESS AND OFFSET

Enter a value for the **Thickness**  $d$  of the selected boundaries. The default is 0.01 m. The thickness can be variable if an expression is used.



Select an option from the **Offset definition** list — **No offset**, **Relative offset**, or **Physical offset**.

- For **No offset**, the modeled boundary coincides with the shell midsurface.
- For **Relative offset**, enter a value or expression in the  $z_{\text{rel\_offset}}$  field for the offset as the ratio between the offset distance and half the shell thickness. A value of +1 means that the actual shell bottom surface is located on the meshed boundary, and a value of -1 means that the shell top surface is located on the meshed boundary.]



Figure 5-6: Meshed boundary indicated in red. The vector  $\mathbf{n}$  indicates the positive orientation of the shell normal.

Values of  $z_{\text{rel\_offset}}$  outside the range  $[-1,1]$  are also allowed.

- For **Physical offset**, enter a value or expression in the  $z_{\text{offset}}$  field for the actual distance from the meshed boundary to the shell midsurface.

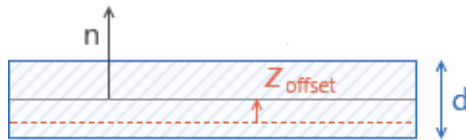


Figure 5-7: Meshed boundary indicated in red. The vector  $\mathbf{n}$  indicates the positive orientation of the shell normal.

	<p>The offset settings are not available for the Plate interface.</p> <p>Thickness and offset settings specified on this node are not applicable for a <b>Layered Linear Elastic Material</b> node as this information is provided on <b>Layered Material</b>, and <b>Layered Material Link</b> nodes.</p>
	<p>For theory, see <a href="#">Offset</a>.</p>

## LOCATION IN USER INTERFACE

### *Context Menus*

**Shell>Thickness and Offset**

**Plate>Thickness and Offset**

### *Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Shell>Thickness and Offset**

Physics tab with **Plate** selected:

**Domains>Plate>Thickness and Offset**

### *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](#)
- [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 a linear elastic **Solid model** — **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  $E_z$ ,  $\nu_{yz}$ , or  $\nu_{xz}$  need to be entered due to the shell assumptions.

- 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 box.

If the **Solve for out-of-plane strain components** check box 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

In this section there is always one check box. Either **Force linear strains** or **Include geometric nonlinearity** is shown.

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are however some cases when you would still want to use a small strain formulation for a certain domain. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.

The default value is that the check box is cleared, except when opening a model created in a version prior to 4.3. In this case the state is chosen so that the properties of the model are conserved.



The **Include geometric nonlinearity** check box is displayed only if the model was created in a version prior to 4.3, and geometric nonlinearity was originally used for the selected domains. It is then selected and forces the **Include geometric nonlinearity** check box in the study step to be selected. If the check box is cleared, it is permanently removed and the study step assumes control over the selection of geometric nonlinearity.

## 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**

### *Layered Linear Elastic Material*

---

The **Layered Linear Elastic Material** 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 **Layered Linear Elastic Material** node you can incorporate many other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)

- External Stress
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Damage
- Safety

Some of these material models are only available together with the Nonlinear Structural Materials Module (see <https://www.comsol.com/products/specifications/>).



The **Layered Linear Elastic Material** 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 5-2: 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 a linear elastic **Solid model** — **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  $E_z$ ,  $\nu_{yz}$ , or  $\nu_{xz}$  need to be entered due to the shell assumptions.
- 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 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**, or **Strain 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 box.

If the **Solve for out-of-plane strain components** check box 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.

#### **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 box.

### **GEOMETRIC NONLINEARITY**

In this section there is one check box **Force linear strains**. If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are however some cases when you would still want to use a small strain formulation for a certain domain. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.

### **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 box.

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Shell>Material Models>Layered Linear Elastic Material**

#### *Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Material Models>Layered Linear Elastic Material**

#### *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**

### *Layered Hyperelastic Material*

The **Layered Hyperelastic Material** 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. This material model requires the Nonlinear Structural Materials Module.

If the Composite Materials Module analysis 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** check box in the study settings is selected and cannot be cleared.

By adding the following subnodes to the **Layered Hyperelastic Material** node you can incorporate many other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Damping](#)
- [Hygroscopic Swelling](#)
- [Viscoelasticity](#)
- [External Stress](#)
- [Mullins Effect](#)
- [External Strain](#)

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](#).

---

By adding the following subnodes to the **Layered Hyperelastic Material** node you can incorporate many other effects. Some of these material models are only available together with the Nonlinear Structural Materials Module (see <https://www.comsol.com/products/specifications/>).



The **Layered Hyperelastic Material** 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 5-3: LAYER SELECTIONS; LINEAR ELASTIC MATERIAL

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default when added.	When <b>Use all layers</b> 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 for the **Hyperelastic Material** node 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 which are part of the selection of a layered material defined in **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 box.

## LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Material Models>Layered Hyperelastic Material**

*Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Material Models>Layered Hyperelastic Material**

*Viscoelasticity*

Use the **Viscoelasticity** subnode to add viscous stress contributions to a [Layered Linear Elastic Material](#) or to a [Layered Hyperelastic Material](#).

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 5-4: LAYER SELECTIONS; VISCOELASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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](#).

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Shell>Layered Linear Elastic Material>Viscoelasticity**

**Shell>Layered Hyperelastic Material>Viscoelasticity**

### *Ribbon*

Physics tab with **Layered Linear Elastic Material** or **Layered 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 the [Layered Linear Elastic Material](#) and requires the Nonlinear Structural Material Module.

If the Composite Materials Module analysis 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 5-5: LAYER SELECTIONS; PLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**Shell>Layered Linear Elastic Material>Plasticity**

### *Ribbon*

Physics tab with **Layered Linear Elastic Material** node selected in the model tree:

**Attributes>Plasticity**

### *Set Variables*

Use the **Set Variables** subnode to [Plasticity](#) of a [Layered 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 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 Model](#) 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>Layered Linear Elastic 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 [Layered Linear Elastic Material](#).

Creep models require the Nonlinear Structural Material Module.

If the Composite Materials Module analysis 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 5-6: LAYER SELECTIONS; CREEP

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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](#).

## LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Layered Linear Elastic Material>Creep**

*Ribbon*

Physics tab with **Layered Linear Elastic Material** node selected in the model tree:

**Attributes>Creep**

*Viscoplasticity*

Use the **Viscoplasticity** subnode to define the viscoplastic properties of a [Layered Linear Elastic Material](#).

The Nonlinear Structural Material Module is required for this material model.

If the Composite Materials Module analysis 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 5-7: LAYER SELECTIONS; VISCOPLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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](#).

## LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Layered Linear Elastic Material>Viscoplasticity**

*Ribbon*

Physics tab with **Layered Linear Elastic Material** node selected in the model tree:



## Attributes>Viscoplasticity

### *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](#), [Layered Linear Elastic Material](#), or [Layered Hyperelastic Material](#). 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 **Layered Linear Elastic Material** or **Layered Hyperelastic Elastic Material** node. In this section, select the layers in which thermal expansion needs to be modeled.

TABLE 5-8: LAYER SELECTIONS; THERMAL EXPANSION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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**  $T_{\text{ref}}$  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/soll)**), 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

Specify the thermal properties that define the thermal strain.

From the **Coefficient of thermal expansion**  $\alpha$  list, select **From material** to use the coefficient of thermal expansion from the material, or **User defined** to enter a value or expression for  $\alpha$ . Select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . When a nonisotropic coefficient of thermal expansion 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** or **Layered Linear Elastic Material**.

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.

### *Linear Elastic Material*

Enter the **Temperature difference in thickness direction**  $\Delta T_z$ . This is the temperature difference between the top and bottom surfaces.

### *Layered Linear Elastic or Layered Hyperelastic Material*

From the list, select **Temperature difference in thickness direction** or **Temperature gradient in thickness direction**.

When **Temperature difference in thickness direction** is selected, enter the temperature difference  $\Delta T_z$  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>Layered Linear Elastic Material>Thermal Expansion**

**Shell>Layered Hyperelastic Material>Thermal Expansion**

**Plate>Linear Elastic Material>Thermal Expansion**

##### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material** or **Layered Hyperelastic 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  $\beta_h$  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 thickness direction of the shell.

Hygroscopic swelling can be modeled for the [Linear Elastic Material](#), [Layered Linear Elastic Material](#), and [Layered Hyperelastic Material](#). 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 **Layered Linear Elastic Material** or **Layered Hyperelastic Material** node. In this section, select the layers in which hygroscopic swelling needs to be modeled.

TABLE 5-9: LAYER SELECTIONS; HYGROSCOPIC SWELLING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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**  $c_{\text{ref}}$ . 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**  $\beta_h$  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  $\beta_h$ . The default value for the **User defined** case is  $1.5e-4 \text{ m}^3/\text{kg}$ . 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** check box 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**, **Layered Linear Elastic Material** or **Layered Hyperelastic Material**.

The concentration is then assumed to vary linearly through the thickness.

#### *Linear Elastic Material*

Enter the **Concentration difference in thickness direction**  $\Delta c_z$ . This is the difference between the concentrations at the top and bottom surfaces.

#### *Layered Linear Elastic or Layered Hyperelastic Material*

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  $\Delta c_z$  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>Layered Linear Elastic Material>Hygroscopic Swelling**

**Shell>Layered Hyperelastic Material>Hygroscopic Swelling**

**Plate>Linear Elastic Material>Hygroscopic Swelling**

#### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material** or **Layered 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](#), [Layered Linear Elastic Material](#), 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 the **Layered Linear Elastic Material**, 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 **Layered Linear Elastic Material** node. Select the layers in which initial stress and strain needs to be modeled.

TABLE 5-10: 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 <b>Use all layers</b> 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 Layered Linear Elastic Material 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**  $\gamma_0$
- **Initial bending strain**  $\chi_0$
- **Initial transverse shear strain**  $\zeta_0$



- 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 Stresses and Strains](#).
- 

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Shell>Linear Elastic Material>Initial Stress and Strain**

**Shell>Layered Linear Elastic Material>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**, **Layered Linear Elastic Material**, 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](#), [Layered Linear Elastic Material](#), [Layered Hyperelastic Material](#), 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 **Layered Linear Elastic Material** or **Layered Hyperelastic Material** node.

Select the layers in which external stress is to be modeled.

TABLE 5-11: LAYER SELECTIONS; EXTERNAL STRESS

<b>SELECTION</b>	<b>USE ALL LAYERS</b>	<b>SELECTION OF INDIVIDUAL LAYERS</b>
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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](#).

*Layered Linear Elastic Material, Layered Hyperelastic Material, or Section Stiffness*  
You can select any available boundary system.

## EXTERNAL STRESS

Select a **Stress input** — **Stress tensor** 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** 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 also the entry **User defined**. When **User defined** is selected, you can enter the data for the membrane (bending, shear) part of the **External stress tensor**  $\sigma_{\text{ext},m}$  ( $\sigma_{\text{ext},b}$ ,  $\sigma_{\text{ext},s}$ ) 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.

Choose a **Contribution type** — **Add to stress tensor** or **Load contribution only** to determine the effect of the contribution.

- When **In-plane force** is selected, you enter the external stress in the form of section forces. Specify the external stress as values or expressions for the **In-plane force**  $N_{\text{ext}}$ , the **Moment**  $M_{\text{ext}}$ , and the **Out-of-plane shear force**  $Q_{\text{ext}}$ . Choose a **Contribution type** — **Add to stress tensor** or **Load contribution only** 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>Layered Linear Elastic Material>External Stress**

**Shell>Layered Hyperelastic Material>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**, **Layered Linear Elastic Material**, **Layered Hyperelastic Material**, 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 [Layered Hyperelastic Material](#).

### 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  $F_{\text{ext}}$ . 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  $F_{\text{ext}}^{-1}$ . 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>Layered Hyperelastic Material>External Strain**

### *Ribbon*

Physics tab with **Layered Hyperelastic Material** node selected in the model tree:

**Attributes>External Strain**

### *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 [Layered Linear Elastic Material](#).

## **SHELL PROPERTIES**

Select the layer or layers for which the damage is to be computed.

TABLE 5-12: LAYER SELECTIONS; DAMAGE

<b>SELECTION</b>	<b>USE ALL LAYERS</b>	<b>SELECTION OF INDIVIDUAL LAYERS</b>
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Damage** node 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>Layered Linear Elastic Material>Damage**

*Ribbon*

Physics tab with **Layered Linear Elastic Material** 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 [Layered Hyperelastic Material](#).

## 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**  $d_{\infty}$ . The default is 1.
- **Damage saturation**  $W_{\text{sat}}$ . The default is 1 MJ/m<sup>3</sup>.
- **Deformation dependence coefficient**  $\alpha$ . The default is 0.

For **Miehe** enter the following settings:

- **Maximum damage**  $d_{\infty}$ . The default is 1.
- Damage saturation**  $W_{\text{sat}}$ . The default is 1 MJ/m<sup>3</sup>.

## LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Layered Hyperelastic Material>Mullins Effect**

*Ribbon*

Physics tab with **Layered Hyperelastic Material** 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 **Layered Linear Elastic Material**. Four different variables describing the failure risk are defined, as described in [Table 5-13](#).

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 5-13: 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 **Layered Linear Elastic Material** node.

Select the layer or layers for which the safety factor variables are to be defined.

TABLE 5-14: LAYER SELECTIONS; SAFETY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**Shell>Linear Elastic Material>Variables>Safety**

**Shell>Layered Linear Elastic Material>Variables>Safety**

**Plate>Linear Elastic Material>Variables>Safety**

### *Ribbon*

Physics tab with **Linear Elastic Material** or **Layered Linear 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 **Layered Linear Elastic Material** node.

Select the layer or layers for which this damping model is to be used.

TABLE 5-15: LAYER SELECTIONS; DAMPING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**Shell>Linear Elastic Material>Damping**

**Shell>Layered Linear Elastic Material>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**, **Layered Linear Elastic Material**, **Layered Hyperelastic Elastic Material**, or **Section Stiffness** node selected in the model tree:

**Attributes>Damping**

## 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:

- [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 unit 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  $\mathbf{D}_{As}$  and  $\mathbf{D}_{as}$ . Is so, set both shear correction factors to  $I$ .

### GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are however some cases when you would still want to use a small strain formulation for a certain domain. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.

## 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 box.



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 an edge, boundary, domain, or point condition to a model where the displacements and rotations are prescribed in one or more directions.

With this condition it is possible to prescribe a displacement in one direction or one of the rotations, leaving the shell free to deform or rotate in the other directions.

- If zero displacements and rotations are prescribed in all directions, this is the same as **Fixed Constraint**.
- If zero displacements are prescribed in all directions, this is the same as **Pinned**.
- If zero rotations are prescribed in all directions, this is the same as **No Rotation**.

## COORDINATE SYSTEM SELECTION

Specify 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

To define a prescribed displacement for each spatial direction ( $x$ ,  $y$ , and  $z$ ), select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** check boxes. Then enter a value or expression for the prescribed displacements  $u_0$ ,  $v_0$ , or  $w_0$ .

### 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 direction. Enter a value or expression for the prescribed rotation  $\theta$  in each row for the local tangential directions **t1** and **t2**. Under **For small strains**, select one or both of the **Free rotation around t1 direction** and **Free rotation around t2 direction** check boxes to remove the constraint for the corresponding rotation component. If unchecked, the rotations are constrained to either the input value or to the default zero rotation. The status of the check boxes has no effect when the 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 case of finite rotations.



In 2D axisymmetry, there is only one input, the **Prescribed rotation** around the out-of-plane direction,  $\theta$ .

---


- **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 box.



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 **Fixed Constraint** to prescribe a deformation of the constrained edge caused by changes in temperature of the surroundings. This makes it possible to reduce the stresses caused by such boundary conditions.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  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 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 **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.

- The **Coefficient of thermal expansion**  $\alpha$ . As a 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.  
For **User defined** select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.
- A value or expression for the **Volume reference temperature**  $T_{\text{ref}}$  that 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*

**Shell>Fixed Constraint>Thermal Expansion**

**Plate>Fixed Constraint>Thermal Expansion**

### *Ribbon*

Physics tab with **Fixed Constraint** 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

Specify 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** check boxes. 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** check boxes 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**

Specify 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** check boxes. 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** check boxes 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 box.



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 box.



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 box.



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.

---

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 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 an **Axis to use as symmetry plane normal**. 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.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 box.



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 box.



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)**

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](#).

---




#### **ANTISYMMETRY (EDGES ONLY)**

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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry and Antisymmetry Boundary Conditions</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## 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.

## FORCE

Enter values or expressions for the components  $(x, y, z)$  of the body load  $\mathbf{F}_V$ .

## MOMENT

Enter values or expressions for the components  $(x, y, z)$  of the moment body load  $\mathbf{M}_L$ .



---

After selecting a **Load type**, the **Load** list 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.

---



- 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.

## FORCE

Select a **Load type** — **Force per unit area**, **Total force**, or **Pressure**.

TABLE 5-16:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit area	$F_A$	N/m <sup>2</sup>	boundaries	3D (x, y, z)
			domains	2D (x, y, z)
Total force	$F_{tot}$	N	boundaries	3D(x, y, z)
			domains	2D(x, y, z)
Pressure	$p$	Pa	boundaries	3D
			domains	2D



- A positive pressure is directed in the negative element normal direction.
- 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.

## MOMENT

Enter values or expressions for the components of the moment face load  $\mathbf{M}_A$ .



- 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 local coordinate system.

#### **COORDINATE SYSTEM SELECTION**

Specify 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**.

---

## FORCE

Select a **Load type** — **Force per unit length** (the default), **Force per unit area**, or **Total Force**. Enter values or expressions for the components ( $x, y, z$ ).

TABLE 5-17:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Force per unit length	$F_L$	N/m	edges	3D
			boundaries	2D
Force per unit area	$F_A$	N/m <sup>2</sup>	edges	3D
			boundaries	2D
Total force	$F_{tot}$	N	edges	3D
			boundaries	2D



After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the Shell interface with another physics interface, it is also possible to choose a predefined load from this list.

## MOMENT

Select a **Load type** to define the moment load — **Moment per unit length** (the default) or **Moment per unit area**. Enter values or expressions for the components ( $x, y, z$ ).

This section is available only in the Shell and Plate interfaces.

TABLE 5-18:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
Moment per unit length	$M_L$	N	edges	3D
			boundaries	2D
Moment per unit area	$M_A$	N·m/m <sup>2</sup>	edges	3D
			boundaries	2D
Total moment	$M_{tot}$	N·m	edges	3D
			boundaries	2D



- 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**.

### **FORCE**

Enter values or expressions for the components  $(x, y, z)$  of the point load  $\mathbf{F}_P$ .



The **Load** list normally only contains **User defined**. When combining the Shell 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  $(x, y, z)$  of the point moment  $\mathbf{M}_P$ .



- 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**

## 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.

### FORCE

Select the **Load type** — **Force per unit length**, **Force per unit area**, or **Total force**. Enter values or expressions for  $\mathbf{F}_L$ ,  $\mathbf{F}_A$ , or  $\mathbf{F}_{tot}$ .

### MOMENT

Select the **Load type** — **Moment per unit length**, **Moment per unit area**, or **Total moment**. Enter values or expressions for  $M_L$ ,  $M_A$ , or  $M_{tot}$ .

---



- 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 unit length values, multiplied by  $2\pi r$ , where  $r$  is the radius of the point.
  - 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**

## *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**

Specify 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 5-19](#) 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 5-19: SPRING TYPES FOR THE SPRING FOUNDATION FEATURE

SPRING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Spring constant per unit volume	$k_V$	$N/(m \cdot m^3)$	domains (Plate), boundaries (Shell)
Total spring constant	$k_{tot}$	$N/m$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Spring constant per unit area	$k_A$	$N/(m \cdot m)^2$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Spring constant per unit length	$k_L$	$N/(m \cdot m)$	boundaries (Plate), edges (Shell)
Spring constant	$k_p$	$N/m$	points (Shell, Plate)
Force per volume as function of extension	$F_V$	$N/m^3$	domains (Plate), boundaries (Shell)
Total force as function of extension	$F_{tot}$	$N$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Force per area as function of extension	$F_A$	$N/m^2$	domains (Plate), boundaries (Shell, Plate), edges (Shell)

TABLE 5-19: SPRING TYPES FOR THE SPRING FOUNDATION FEATURE

SPRING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Force per length as function of extension	$F_L$	N/m	boundaries (Plate), edges (Shell)
Force as function of extension	$F_P$	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**  $\eta_s$  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**  $\eta_k$  or  $\eta_f$  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 5-20](#) 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 5-20: DAMPING TYPES FOR THE SPRING FOUNDATION FEATURE

DAMPING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Damping constant per unit volume	$d_V$	N·s/(m·m <sup>3</sup> )	domains (Plate), boundaries (Shell)
Damping constant per unit area	$d_A$	N·s/(m·m <sup>2</sup> )	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Total damping constant	$d_{tot}$	N·s/m	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Damping constant per unit length	$d_L$	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**  $u_0$ .



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** or **Thin Elastic Layer** 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**

Specify 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 for the **Added Mass** node 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.

The [Point Mass Damping](#) subnode can be added to specify a mass-proportional damping.

## COORDINATE SYSTEM SELECTION

With the **Coordinate system** list, control the coordinate system around the axis of which the principal mass moment of inertias are defined.

### POINT MASS

Enter a **Point mass**  $m$ .

Enter a single value for an isotropic **Mass moment of inertia  $\mathbf{J}$** , or select **Diagonal** or **Symmetric** to enter a full moment of inertia tensor.

### FRAME ACCELERATION FORCES

Click to select the **Exclude contribution** check box to switch off the loads that can be caused by the point mass when the frame is accelerated when using a [Gravity](#) or

[Rotating Frame](#) feature. The setting will also determine whether the node will contribute when [Computing Mass Properties](#).

#### 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**

#### *Point Mass Damping*

---

Use the **Point Mass Damping** subnode to add damping to a [Point Mass](#) parent node.

#### POINT MASS DAMPING

Enter a **Mass damping parameter**  $\alpha_{dM}$ . This is the mass proportional term of a Rayleigh damping.

#### LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Point Mass>Point Mass Damping**

**Plate>Point Mass>Point Mass Damping**

*Ribbon*

Physics tab with **Point Mass** node selected in the model tree:

**Attributes>Point Mass Damping**

#### *Periodic Condition*

---

Use a **Periodic Condition** to prescribe that the displacements and rotations on two different sets of edges with the same geometrical shape are related, as in a periodic structure. In the Plate interface the connection is between boundaries rather than edges as is the case in the Shell interface.

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 both between displacements and between rotations. They 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 modeled structure is typically a unit cell of a repetitive structure.
- The **Cyclic symmetry** is a special case of a Floquet condition, intended for structures which consist of a number of sectors which are identical when rotated around a common axis, like in a fan.

The two sets of edges 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, 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

Select both the source and destination edges.

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, use the **Destination Selection** subnode to 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 geometry edges for the matching to work properly.



In the Plate interface, **Edge Selection** is replaced by **Boundary Selection**.

---

## PERIODICITY SETTINGS

With **Type of periodicity** you select the form of periodicity that your solution should have.

- 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  $\theta_S$ .

---




If any point on the edges having the periodic condition is located on the axis of cyclic symmetry, enter the **Axis direction** vector,  $\mathbf{t}_c$ . This orientation of the axis of cyclic symmetry is then needed for eliminating conflicting constraints.

In the Shell interface, you always must enter a value for the sector angle  $\theta_s$ .

In either case, 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 check box 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 box.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)

## ORIENTATION OF SOURCE

In **Transform to intermediate map**, select a coordinate system representing the orientation of the degrees of freedom on the source selection. The corresponding setting for the destination is given in the **Destination Selection** subnode.



- [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**

**Shell>Connections>Periodic Condition>Destination Selection**

**Plate>Periodic Condition>Destination Selection**

### *Ribbon*

Physics tab with **Shell** selected:

**Edges>Connections>Periodic Condition**

Physics tab with **Plate** selected:

**Boundaries>Periodic Condition**

Physics tab with **Periodic Condition** node selected in the model tree:

**Attributes>Destination Selection**



The **Solid Connection** node is obsolete, and has been superseded by the [Solid-Thin Structure Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Solid Connection** node will be removed in future versions.

---

A shell can be coupled to a solid by adding a **Solid Connection** node in the Shell interface and a [Shell Connection](#) node in the Solid Mechanics interface. The connection can either be from an edge of the shell to a boundary on the solid, or between two boundaries. The first case is intended for modeling a transition from a shell to a solid where shell assumptions are valid on both sides of the connection. The second case is for adding a shell on top of a solid. Select this feature from the **Connections** submenu.

#### **SOLID CONNECTION (EDGES ONLY)**

Select a **Connection type** — **Softened** or **Simplified**. When using **Softened** (the default), three extra degrees of freedom are added to each selected point. This allows for a more accurate description of the transition, but the model can in some cases become underconstrained if the mesh on the solid is very coarse. The **Simplified** version of the connection just adds constraints to the boundary of the solid, and in general causes local disturbances of the stress field.

When using the **Solid Connection** node for a boundary, the connection is intended for placing a shell as a cladding on the boundary of a solid. No other settings except the boundary selection are required.

---



- 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](#).
-

## LOCATION IN USER INTERFACE

*Context Menus*

**Shell>Connections>Solid Connection**

*Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Connections>Solid Connection**

**Edges>Connections>Solid Connection**

*Beam Connection*

---



The **Beam Connection** node is obsolete, and has been superseded by the [Shell-Beam Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Beam Connection** node will be removed in future versions.

---

A shell can be connected to a beam by adding a **Beam Connection** node in the Shell interface and a [Shell Connection](#) node in the Beam interface.

The connection can be between:

- Two edges, one in each interface
  - A point in the Beam interface and a boundary in the Shell interface
  - A point in the Beam interface and an edge in the Shell interface
- 



- For more information about coupling different element types, see [Coupling Techniques](#).
  - For details about the formulation of this coupling, see [Connection Between Shells and Beams](#)
- 

## BEAM CONNECTION (EDGES)

Select a **Connected entity** — **Beam point** or **Beam edge**.



When **Beam point** is selected, select the name of the corresponding **Shell Connection** node defined at the point level in the Beam interface to specify the connected parts.

For **Beam point** select an option from the **Connected region** list — **Selected edges** (the default), **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` in [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.

For **Beam edge** select the name of the corresponding **Shell Connection** node defined at the edge level in the Beam interface to specify the connected parts. Select an **Edge type** — **Shared** (the default) or **Parallel**.

- For **Shared** select an **Offset definition** — **Along shell normal** (the default) or **Offset vector**. For **Along shell normal** enter an **Offset**  $\zeta$ . 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. The **Offset vector** is interpreted in the coordinate system selected in the **Coordinate System Selection** section.
- For **Parallel** 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.

#### **BEAM CONNECTION (BOUNDARIES)**

Select the name of the corresponding **Shell Connection** node defined at the point level in the Beam interface to specify the connected parts.

Select an option from the **Connected region** list — **Selected boundaries** (the default), **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` in 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 edges** option.

## LOCATION IN USER INTERFACE

### *Context Menus*

**Shell>Connections>Beam Connection**

### *Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Connections>Beam Connection**

**Edges>Connections>Beam Connection**

### *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 may 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** check box in the **Advanced** section.

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:

- [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 any additional coordinate systems that the model includes. 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.

## 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** when the main selection is at edge level, select **Entity level** — **Edge** or **Point**. 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 edges to which the rigid connector is attached. As a special case, you can select a single point, and thus locate the center of rotation at a certain point.

- For **Centroid of selected entities**:
  - 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.
- For **User defined**, in the **Global coordinates of center of rotation**  $\mathbf{X}_C$  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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

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 Y**, and **Prescribed in Z direction** check boxes. Then enter a value or expression for the prescribed displacements  $u_0$ ,  $v_0$ , or  $w_0$ .

### PRESCRIBED ROTATION AT CENTER OF ROTATION

Select an option from the **By** list — **Free** (the default), **Constrained rotation**, or **Prescribed rotation at center of rotation**.

- For **Constrained rotation** select one or more of the **Constrain rotation about X**, **Constrain rotation about Y**, and **Constrain rotation about Z axis** check boxes in order to enforce zero rotation about the corresponding axis in the selected coordinate system.
- For **Prescribed rotation at center of rotation** enter an **Axis of rotation**  $\Omega$  and an **Angle of rotation**  $\phi$ . The axis of rotation is given in the selected coordinate system.

### 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.

### 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** check box. The check box, which is checked 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 check box. 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.



- 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 Theory](#)
- [Harmonic Perturbation](#)
- [Load Cases](#)

## LOCATION IN USER INTERFACE

### *Context Menus*

**Shell>Face Constraints>Rigid Connector** (Boundary)

**Shell>Connections>Rigid Connector** (Edge)

**Shell>Points>Rigid Connector** (Point)

### *Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Face Constraints>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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

### **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 **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.
- The **Coefficient of thermal expansion**  $\alpha$ . As a default, values **From material** are used. This requires that a material has been assigned to the boundaries of the rigid connector selection.

For **User defined** select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.

- A value or expression for the **Volume reference Temperature**  $T_{\text{ref}}$  which is the temperature at which there are no thermal displacements in the rigid connector.



- [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.



The **Attachment** node is available with either the Multibody Dynamics Module or the Rotordynamics Module. The use of attachments is discussed in the [Attachments](#) section in the *Multibody Dynamics Module User's Guide*.

---

## 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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

## 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 **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.
- The **Coefficient of thermal expansion**  $\alpha$ . As a default, values **From material** are used. This requires that a material has been assigned to the boundaries of the attachment selection.

For **User defined** select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more

components for a general coefficient of thermal expansion tensor  $\alpha$ . 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.

- A value or expression for the **Volume reference Temperature**  $T_{\text{ref}}$  which is the temperature at which there are no thermal displacements in the attachment.



- [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 ( $\phi$ ), 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  $F_{ph}$  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  $M_{ph}$  for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

## LOCATION IN USER INTERFACE

### *Context Menus*

**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.



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**

# Layered Shell

This chapter describes the Layered Shell interface, which is found under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Theory for the Layered Shell Interface](#)
- [The Layered Shell Interface](#)

# Theory for the Layered Shell Interface

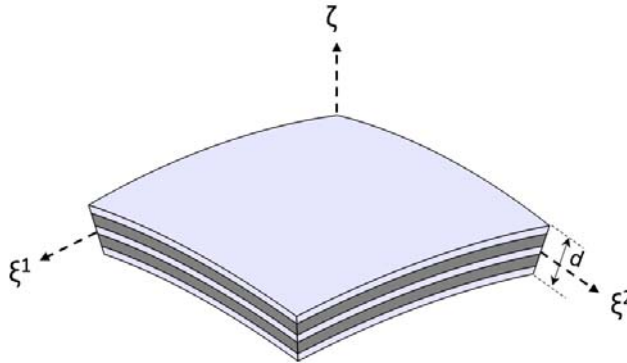
The theory for the Layered Shell interface is discussed in this section:

- [About Layered Shells](#)
- [Theory Background for the Layered Shell Interface](#)

## *About 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 6-1: Geometry of a doubly curved laminated shell.*

Figure 6-1 shows the uniform thickness doubly curved laminated shell having an orthogonal curvilinear coordinate system  $(\xi^1, \xi^2, \zeta)$  and a total thickness  $(d)$ .

A typical stacking sequence of a composite laminate having  $n$  layers is shown in Figure 6-2. The thickness of each layer  $(d_k)$ , as well as the fiber direction in each layer



$(\theta_k)$  with respect to first principal direction  $(\xi^1)$  of the laminate are indicated. A counterclockwise rotation of the fiber direction with respect to  $(\zeta)$  direction is considered as positive.

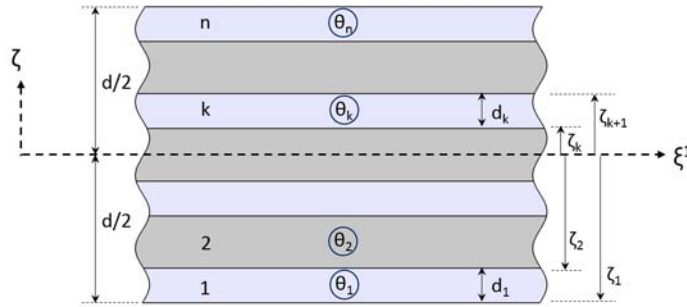


Figure 6-2: A typical stacking sequence of a composite laminate showing thickness and fiber orientation in each layer.



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 **Layered Linear Elastic Material** 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) of modeling a layered shell is described in [Theory for FSDT Laminated Shell](#).

### Theory Background for the Layered Shell Interface

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.

This section covers the following topics:

- [Physics Interface Selection](#)
- [Governing Equations](#)
- [Gradient Definition](#)
- [Area Scale Factor](#)
- [Integrations](#)
- [Discretization](#)
- [Layer Materials, Thicknesses, and Orientations](#)
- [Offset and Local Coordinate System](#)
- [Transform and Scale](#)
- [Fold-Line Connection](#)
- [Layered Shell Continuity](#)
- [Using the Extra Dimension Coordinates](#)
- [Results Evaluation in Layered Shells](#)

#### **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.

### 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  $F$  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

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla_T \mathbf{u} + \frac{d\mathbf{u}}{dz} \mathbf{n}^T + I \quad (6-1)$$

where

- $\nabla_T$  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 6-1 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

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \left( \nabla_T \mathbf{u} + \frac{d\mathbf{u}}{dz} \mathbf{n}^T + I \right) \cdot \left( \frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} \right)^{-1}$$

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} = I + \left( z + \frac{d}{2}(-1 + z_{off}) \right) \nabla_T \mathbf{n}$$

where

- $\mathbf{X}_r$  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

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \left( \nabla_T \mathbf{u} + \frac{1}{lsc} \frac{d\mathbf{u}}{dz} \mathbf{n}^T + I \right) \cdot \left( \frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} \right)^{-1}$$

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} = I + \nabla_T \left( \left( z_s + \frac{d_s}{2}(-1 + z_{off}) \right) \mathbf{n} \right)$$

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 for each layer of the laminate can be defined as:

$$ASF = \left| \frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} \right|$$

### INTEGRATIONS

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:

$$\int_V f(\xi^1, \xi^2, \zeta) dV = \int_A \int_{\zeta} f_1(\xi^1, \xi^2) f_2(\zeta) d\zeta dA = \int \sum_{A_k=1}^n \int_{\zeta^k}^{\zeta^{k+1}} f_1(\xi^1, \xi^2) f_2(\zeta) dA$$

As discussed in the previous section, an 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

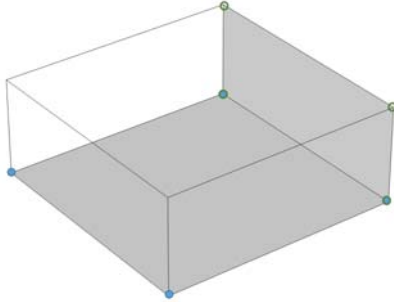
The Layered Shell interface can use different shape orders for the displacement field in the reference surface and in the through-thickness direction. The shape orders used can be divided into categories:

- Same shape order in the reference surface and in the extra dimension (Figure 6-3 and Figure 6-4)
- Different shape orders in the reference surface and in the extra dimension (Figure 6-5)

Based on the above two categories, 9 different elements are available:

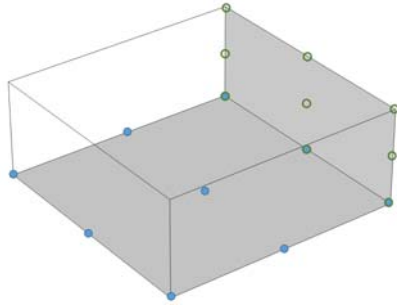
- Linear (8 nodes)
- Quadratic Lagrange (27 nodes)
- Quadratic Serendipity (24 nodes)
- Cubic Lagrange (64 nodes)

- Cubic Serendipity (48 nodes)
- Quadratic-Linear Lagrange (18 nodes)
- Quadratic-Linear Serendipity (16 nodes)
- Quadratic-Cubic Lagrange (36 nodes)
- Quadratic-Cubic Serendipity (32 nodes)

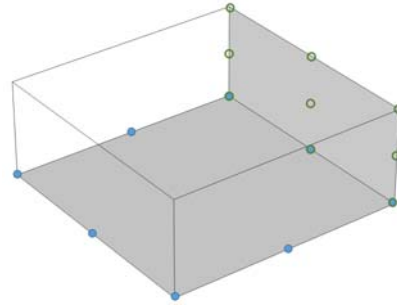


Linear (8 nodes)

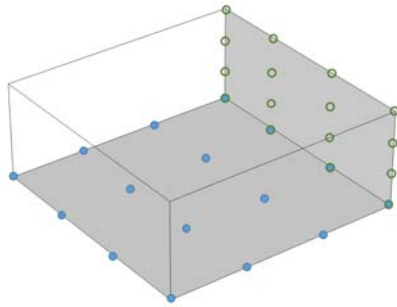
*Figure 6-3: Element having linear shape order.*



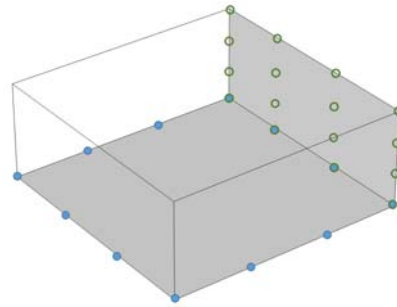
Quadratic Lagrange (27 nodes)



Quadratic Serendipity (24 nodes)

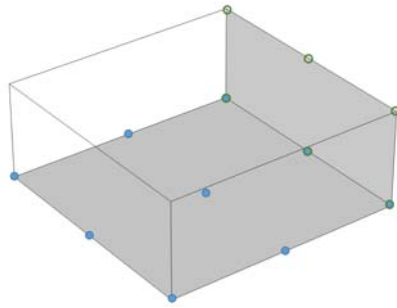


Cubic Lagrange (64 nodes)

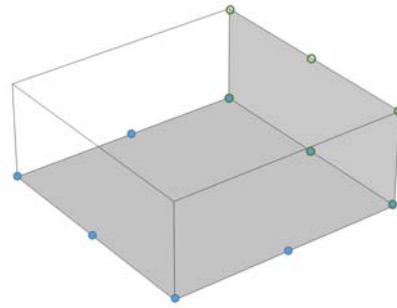


Cubic Serendipity (48 nodes)

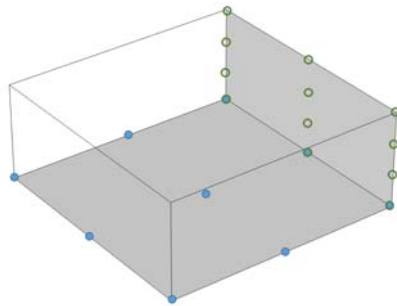
Figure 6-4: Elements having quadratic and cubic shape orders.



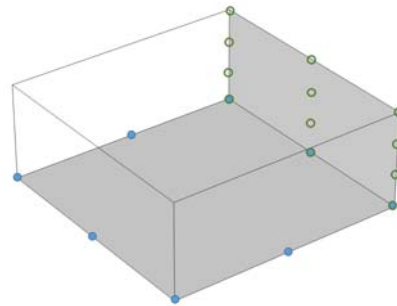
Quadratic-Linear Lagrange (18 nodes)



Quadratic-Linear Serendipity (16 nodes)



Quadratic-Cubic Lagrange (36 nodes)



Quadratic-Cubic Serendipity (32 nodes)

*Figure 6-5: Elements having different shape orders on the reference surface and in the through-thickness direction.*

#### **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.

---

## 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 by the program. 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

$$\mathbf{n}_j \cdot \mathbf{n}_k = \mathbf{u}_{rj} \cdot \mathbf{u}_{rk}$$

$$\mathbf{u}_r = \left( z + \frac{d}{2}(-1 + z_{off}) \right) \mathbf{n} + \mathbf{u} - \mathbf{u}_b$$

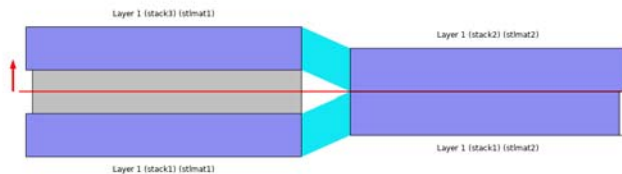
where  $\mathbf{u}_r$  is the displacement vector in the through-thickness direction relative to the displacement vector at the reference surface location.

### LAYERED SHELL CONTINUITY

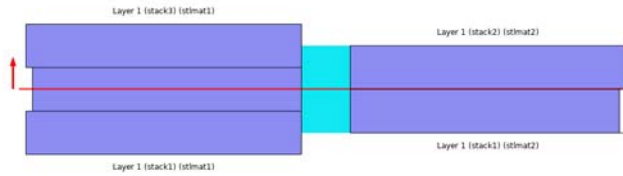
Different layered materials have their own virtual extra dimension geometry and that is why by default two layered materials sharing a common edge in the geometry do not have a continuous displacement field. Thus, when two layered shell surfaces having different layered materials meet side-by-side, a 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:

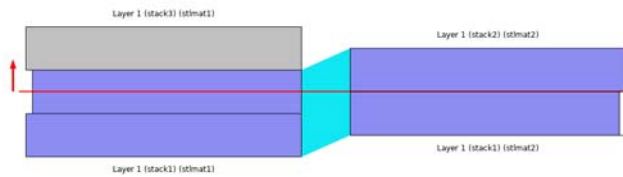
- Connecting the first and third layer of the left layered material to both the layers of the right layered material. This type of connection is typically used in a ply dropping scenario.



- 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.

---

#### 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_1lmat1_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 `1shell.xd` (varies from 0 to the total laminate thickness  $d$ ) and `1shell.xd_rel` (varies from 0 to 1).

Finally, the coordinates in 3D space are available using the physics scoped variables `1shell.X`, `1shell.Y`, and `1shell.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 `1shell.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 etc. 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 of 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.

# 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 **Layered Linear Elastic Material** 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 <https://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 `lshell1`.

## 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** check box is selected. This means that all layers in all layered materials on the selected boundaries are used.

If you deselect the **Use all layers** check box, 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.

## REFERENCE POINT FOR MOMENT COMPUTATION

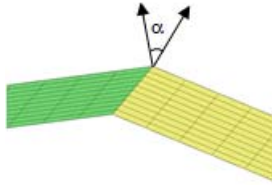
Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

## FOLD-LINE LIMIT ANGLE

The fold-line limit angle  $\alpha$  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  $\alpha$  field. The default



value is 0.05 radians (approximately  $3^\circ$ ). 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](#) for the theory of fold line constraints in layered shell.

#### 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.

#### 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 Stress Analysis 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** check box 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** drop-down 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):

- Activation
- Added Mass
- Added Mass, Interface
- Antisymmetry
- Attachment
- Body Load
- Boundary Load
- Contact
- Continuity
- Coupling Loss
- Creep
- Damage
- Damping
- Delamination
- Dielectric Loss
- Edge Load
- External Stress
- External Strain
- Face Load
- Fixed Constraint
- Fixed Constraint, Interface
- Free
- Gravity
- Hygroscopic Swelling
- Hyperelastic Material
- Initial Values
- Initial Stress and Strain
- Linear Elastic Material
- Line Load
- Mechanical Damping
- Mullins Effect
- Phase
- Piezoelectric Material
- Plasticity
- Point Load
- Predeformation
- Prescribed Acceleration
- Prescribed Acceleration, Interface
- Prescribed Displacement
- Prescribed Displacement, Interface
- Prescribed Velocity
- Prescribed Velocity, Interface
- Rigid Connector
- Rigid Connector, Interface
- Rigid Domain
- Rotating Frame
- Rigid Motion Suppression
- Roller
- Roller, Interface
- Safety
- Set Variables
- Spring Foundation
- Spring Foundation, Interface
- Symmetry
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Thermal Expansion (Rigid Connector)
- Thin Elastic Layer
- Thin Elastic Layer, Interface
- Viscoelasticity
- Viscoplasticity





---

If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence.

---

These nodes are described in the documentation for the Solid Mechanics interface:

- Applied Force (Rigid Connector)
- Applied Force (Rigid Domain)
- Applied Moment (Rigid Connector)
- Applied Moment (Rigid Domain)
- Fixed Constraint (Rigid Domain)
- Initial Values (Rigid Domain)
- Mass and Moment of Inertia (Rigid Connector)
- Mass and Moment of Inertia (Rigid Domain)
- Prescribed Displacement/Rotation
- Spring Foundation (Rigid Domain)



---

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 6-1: 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 for the **Initial Values** node 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
- Damping
- Viscoelasticity
- Plasticity
- Creep
- Viscoplasticity
- Activation
- Safety

### SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 6-2: 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 <b>Use all layers</b> is not selected.  Only a subset of the layers selected in the parent can be selected.

If the elastic 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.

If the properties of the layers differ in the sense of the **Solid model** 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 for the **Linear Elastic Material** node 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. This material model requires the Nonlinear Structural Materials Module.

When a hyperelastic material is included in your model, all studies are geometrically nonlinear. The **Include geometric nonlinearity** check box 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](#)
- [Damping](#)
- [Viscoelasticity](#)
- [Mullins Effect](#)

### SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 6-3: LAYER SELECTIONS; HYPERELASTIC MATERIAL

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When <b>Use all layers</b> 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 for the **Hyperelastic Material** node 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 **Layered Piezoelectric Effect** 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.

This material model requires one of these products: Structural Mechanics Module, MEMS Module, or Acoustics Module.

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 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 **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 **Layered Piezoelectric Effect** 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 [Piezoelectric Material](#) in the *Structural Mechanics Theory* chapter.

### SHELL PROPERTIES

Select the layer or layers for which this material model is to be used.

TABLE 6-4: LAYER SELECTIONS; PIEZOELECTRIC MATERIAL

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When <b>Use all layers</b> 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** ( $\rho$ ).



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

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are, however, some cases when the use of a small strain formulation for a certain domain is needed. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step. The check box is not selected by default to conserve the properties of the model.



- [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 6-5: LAYER SELECTIONS; VISCOELASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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*

**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 6-6: LAYER SELECTIONS; MULLINS EFFECT

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Mullins Effect** node 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](#) and [Hyperelastic Material](#) and requires the Nonlinear Structural Material Module.

## SHELL PROPERTIES

Select the layer or layers for which this plasticity model is to be used.

TABLE 6-7: LAYER SELECTIONS; PLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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](#).
- 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*

Select **Small plastic strains** or **Large plastic strains** to apply either an additive or multiplicative decomposition between elastic and plastic strains.

- When using plasticity together with a hyperelastic material, only the option **Large plastic strains**.

The other 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.

## 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.

- When **Small plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic strain tensor**. 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**.
- When **Large plastic strains** is selected in the **Plasticity Model** list, specify the values for the components of the **Plastic deformation gradient inverse**. From the **Plastic deformation gradient inverse** list, select **Do not set** or **User defined**. The default **User defined** value is the identity 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](#).

Creep models require the Nonlinear Structural Material Module.

### SHELL PROPERTIES

Select the layer or layers for which this creep model is to be used.

TABLE 6-8: LAYER SELECTIONS; CREEP

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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*

**Layered Shell>Linear Elastic Material>Creep**

*Ribbon*

Physics tab with **Linear Elastic Material** node selected in the model tree:

**Attributes>Creep**

### *Viscoplasticity*

---

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model.

The Nonlinear Structural Material Module is required for this material model.

## SHELL PROPERTIES

Select the layer or layers for which this viscoplasticity model is to be used.

TABLE 6-9: LAYER SELECTIONS; VISCOPLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**Layered Shell>Linear Elastic Material>Viscoplasticity**

*Ribbon*

Physics tab with **Linear Elastic Material** node selected in the model tree:

**Attributes>Viscoplasticity**

*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 6-10: LAYER SELECTIONS; THERMAL EXPANSION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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.



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*

**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

$$\varepsilon_{hs} = \beta_h M_m (c_{mo} - c_{mo,ref})$$

where  $\beta_h$  is the coefficient of hygroscopic swelling,  $M_m$  is the molar mass,  $c_{mo}$  is the moisture concentration, and  $c_{mo,ref}$  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 6-11: LAYER SELECTIONS; HYGROSCOPIC SWELLING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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 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 Stresses and Strains](#).

### SHELL PROPERTIES

Select the layer or layers for which this stress or strain contribution is to be added.

TABLE 6-12: 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 <b>Use all layers</b> 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.



- All other settings for the **Initial Stress and Strain** node 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, 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

Select the layer or layers for which this stress contribution is to be added.

TABLE 6-13: LAYER SELECTIONS; EXTERNAL STRESS

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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.



- All other settings for the **External Stress** node 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 stress contribution is to be added.

TABLE 6-14: LAYER SELECTIONS; EXTERNAL STRAIN

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> is not selected.  Only a subset of the layers selected in the parent can be selected.

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.



- All other settings for the **External Strain** node 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**

### *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](#).

## 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 <b>Use all layers</b> 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 for the **Damage** node 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 6-16: LAYER SELECTIONS; ACTIVATION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Activation** node 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 6-17: INTERFACE SELECTIONS; DELAMINATION

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Interior interfaces</b> . The choice <b>All interfaces</b> is equivalent to <b>Interior interfaces</b> .	When <b>Selected Interfaces</b> 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**,  $\sigma_t$ . This is the peak stress in pure tension.
- **Shear strength**,  $\sigma_s$ . 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**,  $\lambda$ .

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**  $\alpha$ . 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**,  $\alpha_0$ . The value determines how normal and shear components are combined into a single scalar criterion for damage initiation.
- **Mode mixity exponent**,  $\alpha_c$ . 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 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**,  $\tau$ .

## 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  $\langle \text{phys} \rangle . E_{\text{equ}} / \langle \text{phys} \rangle . d_{\text{ad}}$ .

## ADVANCED

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

**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 6-18](#)

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 6-18: 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<I	FI>I
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=I
Safety factor, SF	For a linear criterion, this is I/FI.	SF>I	SF<I
Margin of safety, MoS	SF-I	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 6-19: LAYER SELECTIONS; SAFETY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**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 6-20: LAYER SELECTIONS; DAMPING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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*

**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](#) and [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 6-21: LAYER SELECTIONS; MECHANICAL DAMPING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Mechanical Damping** node 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 6-22: LAYER SELECTIONS; COUPLING LOSS

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Coupling Loss** node 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 electrical 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 6-23: LAYER SELECTIONS; DIELECTRIC LOSS

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 for the **Dielectric Loss** node 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>Dielectric Loss**

##### *Ribbon*

Physics tab with **Piezoelectric Material** node selected in the model tree:

*Rigid Domain*

Add the **Rigid Domain** node and select one or more boundaries to make them a rigid body. **Rigid Domain** 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 \(Rigid Domain\)](#)).

You can add functionality to the rigid domain through the following subnodes:

- [Fixed Constraint \(Rigid Domain\)](#) to fully constrain the rigid domain.
- [Prescribed Displacement/Rotation](#) to prescribe the displacement of individual degrees of freedom.
- [Applied Force \(Rigid Domain\)](#) to apply a force in given point.
- [Applied Moment \(Rigid Domain\)](#) to apply a moment.
- [Mass and Moment of Inertia \(Rigid Domain\)](#) to add extra mass and moment of inertia in a given point.
- [Spring Foundation \(Rigid Domain\)](#) 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 6-24: 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 for the **Rigid Domain** node are described in the documentation for [Rigid Domain](#) 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 Domain**

*Ribbon*

Physics tab with **Layered Shell** selected:

**Boundaries>Material Models>Rigid Domain**

### *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 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.

Do not select the same contact pair in more than one **Contact** node. Doing so may produce error messages or unpredictable results.



---

In version 5.5, the contact formulation in the Layered Shell interface has the following limitations:

- Only frictionless contact can be modeled. You cannot add the **Friction**, **Adhesion**, or **Decohesion** subnodes under **Contact**.
  - The contact formulation works best if the laminate is thin and the curvature is not too high. Local deformations caused by the contact pressure, such as thickness change, should be small.
  - The best accuracy is obtained when the **Contact surface** is set to **Top** and the position of the layered material is **Upside on boundary**.
- 



---

All settings for the **Contact** node are described in the documentation for [Contact](#) in the Solid Mechanics interface.

---

## LOCATION IN USER INTERFACE

*Context Menus*

**Layered Shell>Pairs>Contact**

*Ribbon*

Physics tab with **Layered Shell** selected:

**Pairs>Contact**

*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 6-25: 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).

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 6-26: 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 <b>Use all layers</b> is not selected. Only a subset of the layers selected in the physics interface can be selected.

If the same layer is selected in two **Prescribed Displacement** nodes being active on the same boundary, the second definition will override the previous.



- All other settings for the **Prescribed Displacement** node 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).

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 6-27: INTERFACE SELECTIONS; PRESCRIBED DISPLACEMENT, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> is selected. Only exterior interfaces can be selected.
Edge, Point	When added: <b>Top interface.</b>	When <b>Selected Interfaces</b> 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 for the **Prescribed Displacement, Interface** node 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 6-28: LAYER SELECTIONS; PRESCRIBED VELOCITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge	Same as interface selection when the node is added.	No

If the same layer is selected in two **Prescribed Velocity** nodes being active on the same boundary, the second definition will override the previous.



- All other settings for the **Prescribed Velocity** node 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 6-29: INTERFACE SELECTIONS; PRESCRIBED VELOCITY, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces</b> .	When <b>Selected Interfaces</b> 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 for the **Prescribed Velocity, Interface** node 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 6-30: LAYER SELECTIONS; PRESCRIBED ACCELERATION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary, Edge	Same as interface selection when the node is added.	No

If the same layer is selected in two **Prescribed Acceleration** nodes being active on the same boundary, the second definition will override the previous.



- All other settings for the **Prescribed Acceleration** node 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 6-31: INTERFACE SELECTIONS; PRESCRIBED ACCELERATION, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> 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 for the **Prescribed Acceleration, Interface** node 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 6-32: 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 <b>Use all layers</b> is not selected.  Only a subset of the layers selected in the physics interface can be selected.



- All other settings for the **Fixed Constraint** node 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 6-33: INTERFACE SELECTIONS; PRESCRIBED DISPLACEMENT, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces</b> .	When <b>Selected Interfaces</b> is selected. Only exterior interfaces can be selected.
Edge, Point	When added: <b>Top interface</b> .	When <b>Selected Interfaces</b> is selected. Any interface can be selected.



- All other settings for the **Fixed Constraint, Interface** node 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**, **Fixed Constraint, Interface**, **Prescribed Displacement**, 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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

### **SHELL PROPERTIES**

This section is only present when the parent node has layer selection.

TABLE 6-34: LAYER SELECTIONS; THERMAL EXPANSION

<b>SELECTION</b>	<b>USE ALL LAYERS</b>	<b>SELECTION OF INDIVIDUAL LAYERS</b>
Same level as parent	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 6-35: 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 <b>Selected Interfaces</b> is selected. Only a subset of the interfaces selected in the parent can be selected.



- All other settings for the **Thermal Expansion** node 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:

**Attributes>Thermal Expansion**

### *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 6-36: 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 for the **Roller** node 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 6-37: INTERFACE SELECTIONS; PRESCRIBED ACCELERATION, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> is selected. Only exterior interfaces can be selected.



- All other settings for the **Roller** node 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 6-38: 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 for the **Symmetry** node 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 6-39: 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 for the **Antisymmetry** node 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 any 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 which 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 6-40: 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 for the **Rigid Motion Suppression** node 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](#) and [Rotating Frame](#) nodes.

## SHELL PROPERTIES

Select the layer or layers to which this load is to be applied.

TABLE 6-41: LAYER SELECTIONS; BODY LOAD

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	When <b>Use all layers</b> is not selected. Only a subset of the layers selected in the interface can be selected.

Contributions from several **Body Load** nodes having a common selection will be added to each other.



- All other settings for the **Body Load** node 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 6-42: INTERFACE SELECTIONS; FACE LOAD

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> 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](#).

### FORCE

Select a **Load type** — **Force per unit area**, **Total force**, or **Pressure**.

Select a **Load type** — **Force per unit area**, **Pressure**, **Total force**, or for 2D components, **Force per unit length**. Then enter values or expressions for the components in the matrix based on the selection and the space dimension.

- For **Force per unit 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 unit 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 6-43:

LOAD TYPE	VARIABLE	SI UNIT
Force per unit area	$F_A$	$\text{N/m}^2$

TABLE 6-43:

LOAD TYPE	VARIABLE	SI UNIT
Total force	$\mathbf{F}_{\text{tot}}$	N
Pressure	$p$	Pa



- 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**

### *Gravity*

When you add a **Gravity** node, gravity 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.

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). The action of gravity can also be seen as a linearly accelerated frame of reference.



## SHELL PROPERTIES

As a gravitational load 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 6-44: LAYER SELECTIONS; GRAVITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as interface selection when the node is added.	No



- All other settings for the **Gravity** node are described in the documentation for [Gravity](#) 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>Gravity**

### *Ribbon*

Physics tab with **Layered Shell** selected:

**Boundaries>Face and Volume Loads>Gravity**

### *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 6-45: 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 for the **Rotating Frame** node 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**

### *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 6-46: LAYER SELECTIONS; BOUNDARY LOAD

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	When <b>Use all layers</b> 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 for the **Boundary Load** node 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 6-47: INTERFACE SELECTIONS; EDGE LOAD

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Edge	When added: <b>Top interface.</b>	When <b>Selected Interfaces</b> 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 for the **Edge Load** node 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 on the layers on a point the layered shell. The load is defined in the given local coordinate system. A point represents an edge going through the thickness of the shell.

### SHELL PROPERTIES

Select the layer or layers to which this load is to be applied.

TABLE 6-48: LAYER SELECTIONS; LINE LOAD

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Point	Same as interface selection when the node is added.	When <b>Use all layers</b> 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 for the **Line Load** node 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 6-49: INTERFACE SELECTIONS; POINT LOAD

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Point	When added: <b>Top interface.</b>	When <b>Selected Interfaces</b> 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 for the **Point Load** node 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 ( $\phi$ ), 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,  $f$ :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

### **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 for the **Phase** node 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 6-50: 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 <b>Use all layers</b> is not selected.  Only a subset of the layers selected in the physics interface can be selected.



- All other settings for the **Spring Foundation** node 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 6-51: INTERFACE SELECTIONS; SPRING FOUNDATION, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> is selected. Only exterior interfaces can be selected.
Edge, Point	When added: <b>Top interface.</b>	When <b>Selected Interfaces</b> is selected. Any interface can be selected.



- All other settings for the **Spring Foundation, Interface** node 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.

#### **SHELL PROPERTIES**

Select the material layers to which the spring condition is to be applied.

TABLE 6-52: LAYER SELECTIONS; THIN ELASTIC LAYER

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	When <b>Use all layers</b> is not selected. Only a subset of the layers selected in the physics interface can be selected.



- All other settings for the **Thin Elastic Layer** node are described in the documentation for [Thin Elastic Layer](#) 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>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 edges 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 6-53: INTERFACE SELECTIONS; THIN ELASTIC LAYER, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Interior interfaces.</b>	When <b>Selected Interfaces</b> is selected. Only interior interfaces can be selected.



- All other settings for the **Thin Elastic Layer, Interface** node are described in the documentation for [Thin Elastic Layer](#) 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>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 for the **Predeformation** node 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>Thin 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 6-54: 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 <b>Use all layers</b> is not selected. Only a subset of the layers selected in the physics interface can be selected.



- All other settings for the **Added Mass** node 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 6-55: INTERFACE SELECTIONS; ADDED MASS, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary, Edge, Point	When added: <b>Top interface.</b>	When <b>Selected Interfaces</b> is selected. Any interface can be selected.



- All other settings for the **Added Mass, Interface** node 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**

### *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.

- [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 6-56: 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 for the **Rigid Connector** node 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.

- [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 6-57: INTERFACE SELECTIONS; RIGID CONNECTOR, INTERFACE

SELECTION	APPLY TO	SELECTION OF INDIVIDUAL INTERFACES
Boundary	When added: <b>Exterior interfaces.</b>	When <b>Selected Interfaces</b> is selected. Only exterior interfaces can be selected.



- All other settings for the **Rigid Connector, Interface** node 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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

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

#### **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 for the **Thermal Expansion** node 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 6-58: LAYER SELECTIONS; ATTACHMENT

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Edge	Same as interface selection when the node is added.	No



The **Attachment** node is available with the Multibody Dynamics Module. The use of attachments is discussed in the [Attachments](#) section in the *Multibody Dynamics Module User's Guide*.

## 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.

The thermal strain depends on the coefficient of thermal expansion  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

### 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 6-59: 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 for the **Thermal Expansion** node 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 box.



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 box.

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**





# Membrane

This chapter describes the **Membrane (mbrn)** interface, which is found under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Theory for the Membrane Interface](#)
- [The Membrane Interface](#)

# Theory for the Membrane Interface

The theory for the Membrane interface is described in this section:

- [About Membranes](#)
- [Theory Background for the Membrane Interface](#)

## *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.

[Theory for the Membrane Interface](#) supports the same study types as the Solid Mechanics interface except it does not include the Linear Buckling study type.

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.

### **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.

#### MEMBRANES FOR 3D MODELS

The Membrane interface in 3D can be active on internal and external 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  $u_{1n}$  and  $u_{2n}$  are additionally added as dependent variables.

#### MEMBRANES FOR 2D AXISYMMETRIC MODELS

The Membrane interface for 2D axisymmetric models can be active on internal and external 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  $u_{1n}$  and  $u_{2n}$  are additionally added as dependent variables.

#### LAYERED AND NONLAYERED MEMBRANES

The Membrane interface has two fundamental material models, [Linear Elastic Material](#) and [Layered Linear Elastic Material](#). In either case, the dependent variables are the same, and exist only on the reference surface. The fundamental difference is that in the **Linear Elastic Material**, the material properties are assumed to be constant through the thickness.

In the **Layered Linear Elastic Material** model, 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.

When the Composite Materials Module is available, the **Layered Linear Elastic Material** model can be used to model multilayered membrane. This is the main use of this material model. It is, however, also used for the [Layered Thermal Expansion](#) 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 important 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.

### *Theory Background for the Membrane Interface*

---

A 3D membrane is similar to a shell but it has only translational degrees of freedom and the results are 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  $(t_1, t_2, n)$ .

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  $\mathbf{x}_1$ ,  $\mathbf{y}_1$ , and  $\mathbf{z}_1$  are defined so that:

- The first direction ( $\mathbf{x}_1$ ) is along the edge. This direction can be visualized by selecting the **Show edge directions arrows** check box in the **View** settings.
- The third direction ( $\mathbf{z}_1$ ) is the same as the membrane normal direction of the adjacent boundary.
- The second direction ( $\mathbf{y}_1$ ) is in the plane of the shell and orthogonal to the edge. It is formed by the cross product of  $\mathbf{z}_1$  and  $\mathbf{x}_1$ ;  $\mathbf{y}_1 = \mathbf{z}_1 \times \mathbf{x}_1$ .

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  $\mathbf{z}_1$  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  $\mathbf{F}$  is in general defined as the gradient of the current coordinates with respect to the original coordinates:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$$

In the Membrane interface, a tangential deformation gradient is computed as

$$\mathbf{F}_T = \mathbf{I} - \mathbf{N} \cdot \mathbf{N}^t + \nabla_T \mathbf{u}$$

Here  $\nabla_T \mathbf{u}$  is a displacement gradient computed using the tangential derivative operator, and  $\mathbf{N}$  is the normal vector to the undeformed membrane.  $\mathbf{F}_T$  now contains information about the stretching in the plane of the membrane.

Since the tangential deformation gradient does not contain any information about the transversal stretch  $\lambda_n$ , it must be augmented by the normal deformation gradient  $\mathbf{F}_N$  to define the full deformation gradient tensor. It is given by

$$\mathbf{F}_N = \lambda_n \mathbf{n} \cdot \mathbf{N}^t = (1 + u_{nn}) \mathbf{n} \cdot \mathbf{N}^t$$

where  $\mathbf{n}$  is the normal vector to the deformed membrane. For anisotropic materials, the shear deformation gradient  $\mathbf{F}_S$  is also necessary to define the full deformation gradient tensor. It is given by

$$\mathbf{F}_S = (u_{1n} \mathbf{t}_1 + u_{2n} \mathbf{t}_2) \cdot \mathbf{N}^t$$

where  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are the tangential vectors to the deformed membrane. The full deformation gradient tensor  $\mathbf{F}$  is the sum of tangential, shear and normal deformation gradient tensors

$$\mathbf{F} = \mathbf{F}_T + \mathbf{F}_S + \mathbf{F}_N = \mathbf{I} - \mathbf{N} \cdot \mathbf{N}^t + \nabla_T \mathbf{u} + (u_{1n} \mathbf{t}_1 + u_{2n} \mathbf{t}_2) \cdot \mathbf{N}^t + (1 + u_{nn}) \mathbf{n} \cdot \mathbf{N}^t$$

Note that  $\mathbf{F}_S$  is only nonzero for anisotropic materials, otherwise  $\mathbf{F}_S = \mathbf{0}$ .

The Right Cauchy-Green tensor  $\mathbf{C}$  is generally defined as

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}$$

From  $\mathbf{C}$ , the Green-Lagrange strains are computed using the standard expression

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$$

The local tangential strains in the membrane are calculated by transformation of this strain 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

$$J = \det(\mathbf{F})$$

In the membrane, only the  $\mathbf{C}$  tensor is available, so instead the following expression is used:

$$J = \sqrt{\det(\mathbf{C})}$$

The area scale factor is also computed as

$$J_A = \frac{J}{1 + u_{nn}}$$

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.

---

### EXTERNAL LOADS

Contributions to the virtual work from the external load are of the form

$$\mathbf{u}_{\text{test}} \cdot \mathbf{F}$$

where the forces ( $\mathbf{F}$ ) can be distributed over a boundary or an edge or be concentrated in a point. In the special case of a follower load, defined by its pressure  $p$ , the force intensity is  $\mathbf{F} = -p\mathbf{n}$  where  $\mathbf{n}$  is the normal 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 spatial frame.

### STRESS CALCULATIONS

The stresses are computed by applying the constitutive law to the computed strains.



The membrane does not support transverse and bending forces so the only section forces it support is the membrane force defined as:

$$N = d \cdot \bar{s}$$

where  $\bar{s}$  is the local stress tensor and contains only in-plane stress components.



# 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 <https://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 of 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.

### **REFERENCE POINT FOR MOMENT COMPUTATION**

Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

### **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 SUBMENU

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):

- [Antisymmetry](#)
- [Creep](#)
- [Damping](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [Layered Linear Elastic Material](#)
- [Linear Elastic Material](#)
- [Safety](#)
- [Symmetry](#)
- [Thickness and Offset](#)
- [Viscoelasticity](#)
- [Viscoplasticity](#)

These nodes are described for the Solid Mechanics interface:

- Added Mass
- Adhesion
- Average Rotation
- Body Load<sup>1</sup>
- Contact
- Creep
- Damping
- Edge Load<sup>3</sup>
- External Stress-Strain Relation
- Face Load<sup>3</sup>
- Fixed Constraint
- Free
- Friction
- Gravity<sup>1</sup>
- Hygroscopic Swelling
- Hyperelastic Material
- Initial Values
- Nonlinear Elastic Material
- Phase
- Plasticity
- Point Load
- Point Load (on Axis)
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement<sup>2</sup>
- Prescribed Velocity
- Ring Load
- Rotating Frame<sup>1</sup>
- Spring Foundation
- Spring-Damper
- Symmetry Plane
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Viscoelasticity
- Viscoplasticity

<sup>1</sup> This is selected from the **Face and Volume Loads** submenu for this interface.

<sup>2</sup> At the boundary level, this is selected from the **Face Constraints** submenu for this interface.

<sup>3</sup> 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.

---

## Thickness and Offset

Use the **Thickness and Offset** node to define the thickness for the membrane. 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 membrane is not located 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 membrane is displaced from the meshed boundary in the direction of the positive boundary normal.

### THICKNESS AND OFFSET

Enter a value for the **Thickness**  $d$  of the selected boundaries. The default is 0.0001 m. The thickness can be variable if an expression is used.

Select an option from the **Offset definition** list — **No offset**, **Relative offset**, or **Physical offset**.

- For **No offset**, the modeled boundary coincides with the membrane.
- For **Relative offset**, enter a value or expression in the  $z_{\text{rel\_offset}}$  field for the offset as the ratio between the offset distance and half the membrane thickness. A value of +1 means that the actual membrane bottom surface is located on the meshed boundary, and a value of -1 means that the membrane top surface is located on the meshed boundary.]



Figure 7-1: Meshed boundary indicated in red. The vector  $\mathbf{n}$  indicates the positive orientation of the boundary normal.

Values of  $z_{\text{rel\_offset}}$  outside the range  $[-1,1]$  are also allowed.

- For **Physical offset**, enter a value or expression in the  $z_{\text{offset}}$  field for the actual distance from the meshed boundary to the membrane.

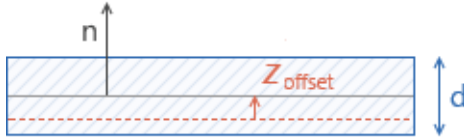


Figure 7-2: Meshed boundary indicated in red. The vector  $\mathbf{n}$  indicates the positive orientation of the boundary normal.



Thickness and offset settings specified on this node are not applicable for a **Layered Linear Elastic Material** node as this information is provided on **Layered Material**, and **Layered Material Link** nodes.

## LOCATION IN USER INTERFACE

*Context Menus*

**Membrane>Thickness and Offset**

*Ribbon*

Physics tab with **Membrane** selected:

**Boundaries>Membrane>Thickness and Offset**

## *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](#)
- [Damping](#)
- [Viscoelasticity](#)

- Plasticity
- Creep
- Viscoplasticity

Note: Some options are only available with certain COMSOL products (see <https://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

To use a mixed formulation by adding the pressure as an extra dependent variable to solve for, select the **Nearly incompressible material** check box. For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem.

Define the **Solid model** and the linear elastic material properties.

#### *Solid Model*

To use a mixed formulation by adding the pressure as an extra dependent variable to solve for, select the **Nearly incompressible material** check box.

Select a linear elastic **Solid model** — **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, so that its stiffness depends on the properties  $E_i$ ,  $\nu_{ij}$ , and  $G_{ij}$ .
- **Anisotropic** for a linear elastic material that has different material properties in different directions, and the stiffness comes from the symmetric elasticity matrix,  $D$ .



- [Material Models](#)
  - [Linear Elastic Material](#)
  - [Orthotropic and Anisotropic Materials](#)
-



### Density

The default **Density**  $\rho$  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 Solid model**, 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**  $\nu$ .
- **Shear modulus**  $G$ .
- **Bulk modulus**  $K$ .
- **Lamé parameter**  $\lambda$  and **Lamé parameter**  $\mu$ .
- **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 **Solid model** list, the material properties vary in orthogonal directions only. 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**  $\nu$ , and the **Shear modulus**  $\mathbf{G}$ . This

defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$



$v_{ij}$  is defined differently depending on the application field. It is easy to transform among definitions, but check which one the material uses.

---

#### *Specification of Elastic Properties for Anisotropic Materials*

When **Anisotropic** is selected from the **Solid model** list, the material properties vary in all directions, and the stiffness comes from the symmetric **Elasticity matrix, D**. 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.

#### **GEOMETRIC NONLINEARITY**


If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all boundaries. There are, however, some rare cases when the use of a small strain formulation for a certain boundary is needed.

In such cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.



- [Modeling Geometric Nonlinearity](#)
  - [Studies and Solvers](#) 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 box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity** or **Viscoelasticity**.

#### **LOCATION IN USER INTERFACE**

*Context Menus*

**Membrane>Material Models>Linear Elastic Material**

*Ribbon*

Physics tab with **Membrane** selected:

**Boundaries>Material Models>Linear Elastic Material**

### *Layered Linear Elastic Material*

---

The **Layered Linear Elastic Material** 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 **Layered Linear Elastic Material** node you can incorporate many other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [Damping](#)
- [Viscoelasticity](#)
- [Plasticity](#)
- [Creep](#)
- [Viscoplasticity](#)
- [Safety](#)

Some of these material models are only available together with the Nonlinear Structural Materials Module (see <https://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 7-1: 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 a linear elastic **Solid model** — **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  $E_z$ ,  $\nu_{yz}$ , or  $\nu_{xz}$  need to be entered due to the membrane assumptions.
- 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**, or **Strain formulation**.

### GEOMETRIC NONLINEARITY

In this section there is one check box **Force linear strains**. If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation in all domains. There are however some cases when you would still want to use a small strain formulation for a certain domain. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step.

When a geometrically nonlinear formulation is used, the elastic deformations used for computing the stresses can be obtained in two different ways if inelastic deformations are present: additive decomposition and multiplicative decomposition. The default is to use multiplicative decomposition. Select **Additive strain decomposition** to change to an assumption of additivity.

### 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 box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by **Creep**, **Plasticity**, **Viscoplasticity** or **Viscoelasticity**.

### LOCATION IN USER INTERFACE

*Context Menus*

**Membrane>Material Models>Layered Linear Elastic Material**

*Ribbon*

Physics tab with **Membrane** selected:

**Boundaries>Material Models>Layered Linear Elastic Material**

## Viscoelasticity

---

Use the **Viscoelasticity** subnode to add viscous stress contributions to an elastic material model. Viscoelasticity can be added to [Linear Elastic Material](#), [Layered Linear Elastic Material](#), [Nonlinear Elastic Material](#) and [Hyperelastic Material](#).

If the Composite Materials Module analysis 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 [Layered Linear Elastic Material](#).

Select the layer or layers for which this viscoelastic model is to be used.

TABLE 7-2: LAYER SELECTIONS; VISCOELASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Viscoelasticity**

**Membrane>Nonlinear Elastic Material>Viscoelasticity**

**Membrane>Hyperelastic Material>Viscoelasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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 either a [Linear Elastic Material](#), [Nonlinear Elastic Material](#), or a [Layered Linear Elastic Material](#); and it requires the Nonlinear Structural Material Module.

If the Composite Materials Module analysis 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 [Layered Linear Elastic Material](#).

Select the layer or layers for which this plasticity model is to be used.

TABLE 7-3: LAYER SELECTIONS; PLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Plasticity**

**Membrane>Nonlinear Elastic Material>Plasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

**Attributes>Plasticity**

### *Set Variables*

Use the **Set Variables** subnode to **Plasticity** of a [Linear Elastic Material](#), [Layered Linear Elastic Material](#), or [Nonlinear Elastic 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>Layered Linear Elastic Material>Plasticity>Set Variables**

**Membrane>Nonlinear Elastic 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](#), [Layered Linear Elastic Material](#), and [Nonlinear Elastic Material](#).

The Nonlinear Structural Material Module is required for this material model.

If the Composite Materials Module analysis is available, the creep 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 **Creep** node is used under [Layered Linear Elastic Material](#).

Select the layer or layers for which this plasticity model is to be used.

TABLE 7-4: LAYER SELECTIONS; CREEP

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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 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>Layered Linear Elastic Material>Creep**

**Membrane>Nonlinear Elastic Material>Creep**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

**Attributes>Creep**

## Viscoplasticity

---

Use the **Viscoplasticity** subnode to define the viscoplastic properties of the material model. Viscoplasticity can be added to [Linear Elastic Material](#), [Layered Linear Elastic Material](#), and [Nonlinear Elastic Material](#).

The Nonlinear Structural Material Module is required for this material model.

If the Composite Materials Module analysis 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 only when the **Viscoplasticity** node is used under [Layered Linear Elastic Material](#).

Select the layer or layers for which this plasticity model is to be used.

TABLE 7-5: LAYER SELECTIONS; VISCOPLASTICITY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Viscoplasticity**

**Membrane>Nonlinear Elastic Material>Viscoplasticity**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, or **Nonlinear Elastic Material** node selected in the model tree:

**Attributes>Viscoplasticity**

### *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](#), [Layered Linear Elastic Material](#), [Nonlinear Elastic Material](#), and [Hyperelastic Material](#). For the **Layered Linear Elastic Material**, 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 [Layered Linear Elastic Material](#).

Select the layer or layers for which this thermal strain contribution is to be added.

TABLE 7-6: LAYER SELECTIONS; THERMAL EXPANSION

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Thermal Expansion**

**Membrane>Nonlinear Elastic Material>Thermal Expansion**

**Membrane>Hyperelastic Material>Thermal Expansion**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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 modeled for [Linear Elastic Material](#), [Layered Linear Elastic Material](#), [Nonlinear Elastic Material](#), and [Hyperelastic Material](#). For the **Layered Linear Elastic Material**, the 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 [Layered Linear Elastic Material](#).

Select the layer or layers for which this swelling strain contribution is to be added.

TABLE 7-7: LAYER SELECTIONS; HYGROSCOPIC SWELLING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Hygroscopic Swelling**

**Membrane>Nonlinear Elastic Material>Hygroscopic Swelling**

**Membrane>Hyperelastic Material>Hygroscopic Swelling**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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](#), [Layered Linear Elastic Material](#), and [Nonlinear Elastic Material](#). For the **Layered Linear Elastic Material**, 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 [Layered Linear Elastic Material](#).

Select the layer or layers for which this stress or strain contribution is to be added.

TABLE 7-8: 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 <b>Use all layers</b> 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**  $\varepsilon_0$  (dimensionless). In addition, you can also provide the **Initial out-of-plane strain**,  $\varepsilon_{n0}$ .

If you know the stress, rather than the force per unit 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 Stresses and Strains](#).
- 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>Layered Linear Elastic Material>Initial Stress and Strain**

**Membrane>Nonlinear Elastic Material>Initial Stress and Strain**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material** 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](#), [Layered Linear Elastic Material](#), [Nonlinear Elastic Material](#), and [Hyperelastic Material](#). For the **Layered Linear Elastic Material**, 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 **Layered Linear Elastic Material** node. Select the layers in which initial stress and strain needs to be modeled.

TABLE 7-9: LAYER SELECTIONS; EXTERNAL STRESS

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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**  $S_{\text{ext}}$  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** or **Load contribution only** 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**  $\sigma_{\text{ext}}$  as **Isotropic**, **Diagonal**, or **Symmetric**. Choose a **Contribution**

**type** — **Add to stress tensor** or **Load contribution only** 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**  $N_{\text{ext}}$ , 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** or **Load contribution only** 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>Nonlinear Elastic Material>External Stress**

**Membrane>Hyperelastic Material>External Stress**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model tree:

**Attributes>External Stress**

### *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](#), [Layered Linear Elastic Material](#), and [Nonlinear Elastic Material](#). Four different variables describing the failure risk are defined, as described in [Table 7-10](#).

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 7-10: 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<I	FI>I
Damage index, DI	A binary value, indicating whether failure is predicted or not. DI is based on the value of FI.	DI=0	DI=I
Safety factor, SF	For a linear criterion, this is I/FI.	SF>I	SF<I
Margin of safety, MoS	SF-I	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 **Layered Linear Elastic Material** node.

Select the layer or layers for which the safety factor variables are to be defined.

TABLE 7-11: LAYER SELECTIONS; SAFETY

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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**  
**Shell>Layered Linear Elastic Material>Variables>Safety**  
**Shell>Nonlinear Elastic Material>Variables>Safety**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, 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 **Layered Linear Elastic Material** node.

Select the layer or layers for which this damping model is to be used.

TABLE 7-12: LAYER SELECTIONS; DAMPING

SELECTION	USE ALL LAYERS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Same as parent selection when the node is added.	When <b>Use all layers</b> 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>Layered Linear Elastic Material>Damping**

**Membrane>Nonlinear Elastic Material>Damping**

**Membrane>Hyperelastic Material>Damping**

### *Ribbon*

Physics tab with **Linear Elastic Material**, **Layered Linear Elastic Material**, **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 box.



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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry Constraints</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

### LOCATION IN USER INTERFACE

*Context Menus*

**Membrane>More Constraints>Antisymmetry**


*Ribbon*

Physics tab with **Membrane** selected:

**Edges>More Constraints>Antisymmetry**



# Beam

This chapter describes the Beam interface, which you find under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Modeling with Beams](#)
- [Theory for the Beam Interface](#)
- [The Beam Interface](#)

# Modeling with Beams

The [Beam Interface](#) theory is described 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.

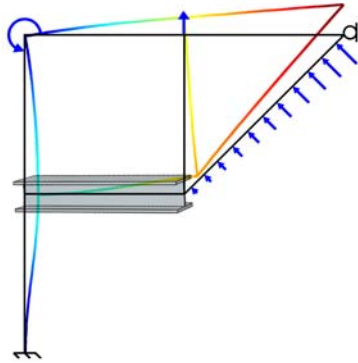
Beams can sustain forces and moments in any direction, both distributed and on individual nodes. The beam's ends and interconnections can be free, simply supported, or clamped. In fact, the simplified boundary conditions are usually responsible for most of the difference that can be found between a beam 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.



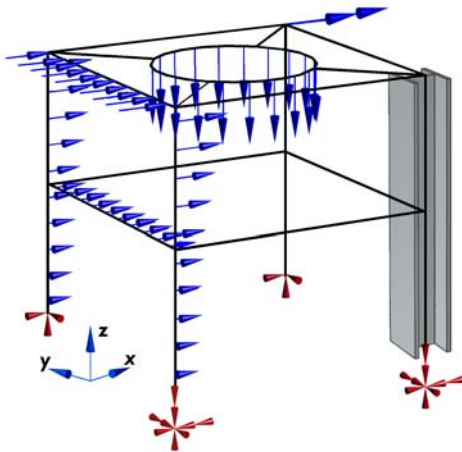
In-plane beams are defined on edges in 2D. They can be used separately or as stiffeners to 2D solid elements.

#### 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  $\theta$  about the global  $z$ -axis.

#### *3D Beam*

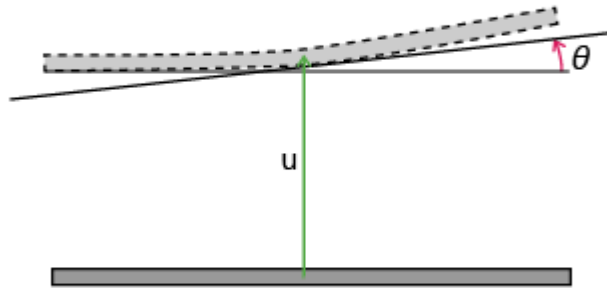
Use the Beam interface in 3D to model three-dimensional frameworks of uniaxial beams.



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  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$  about the global  $x$ -,  $y$ -, and  $z$ -axes.



*Figure 8-1: 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 the stiffness which 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 y-z 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 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 8-1](#) contains a summary of the important geometric variables used in the Beam interface.

TABLE 8-1: 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
$\mu_y$	beam.muy	Max shear stress factor in local y direction
$\kappa_y$	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
$\mu_z$	beam.muz	Max shear stress factor in local z direction
$\kappa_z$	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 Plot Settings Windows](#).

---

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.

### *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. The latter method is more convenient, 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). Then

## Nonrigid Joints

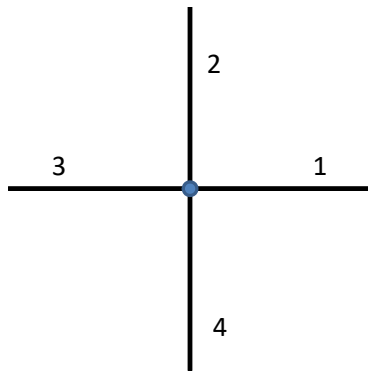
---

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 8-2](#). To model the hinge, the out-of-plane rotation is selected as a decoupled degree of freedom in the **Beam End Release** node.



*Figure 8-2: Original configuration*

There are several ways in which a hinge at the common connection point can be interpreted. Some examples are shown in [Figure 8-3](#) to [Figure 8-5](#). The default behavior is that all beams are disconnected in terms of rotation, as displayed in [Figure 8-3](#).

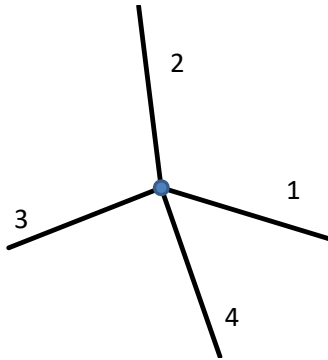


Figure 8-3: All beams are free to rotate relative to each other.

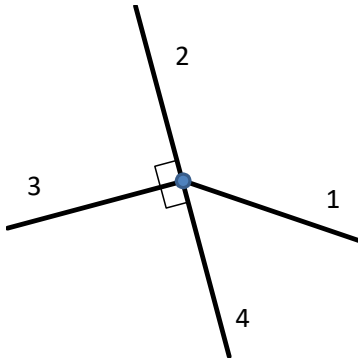


Figure 8-4: Beam 1 is free to rotate with respect to the other three beams.

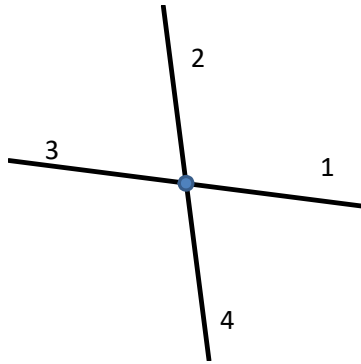


Figure 8-5: Beams 1 and 3 rotate as a common part with respect to beams 2 and 4.

If you want to describe non-default 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 8-4, 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 8-5, 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.



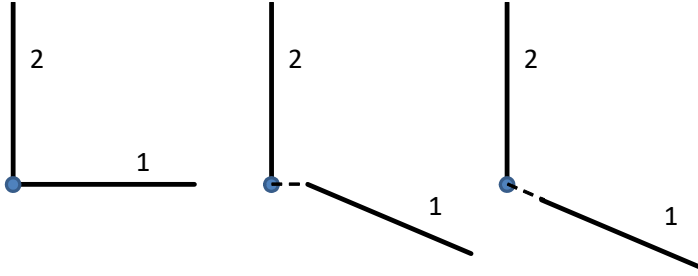
*Stress Analysis of a Portal Crane:* Application Library path  
**Structural\_Mechanics\_Module/Beams\_and\_Shells/portal\_crane**

### 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 8-6. 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 8-6: From left to right: Original configuration, deformed with  $X$  interpreted from beam 2, and deformed with  $X$  interpreted from beam 1.*



# Theory for the Beam Interface

The [Beam Interface](#) theory is described in this section:

- [Shape Functions](#)
- [Geometric Nonlinearity](#)
- [Strain-Displacement/Rotation Relation](#)
- [Stress-Strain Relation](#)
- [Thermal Strain](#)
- [Hygroscopic Swelling](#)
- [Initial Load and Strain](#)
- [Implementation](#)
- [Theory for Section Stiffness](#)
- [Stress Evaluation](#)
- [Limited Displacement](#)
- [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,  $\theta$ , are interpolated as

$$\begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\theta} \end{Bmatrix} = [\mathbf{N}] \begin{Bmatrix} \mathbf{u}_1 \\ \theta_1 \\ \mathbf{u}_2 \\ \theta_2 \end{Bmatrix}$$

where the subscript refers to the two nodes of the element, and  $\mathbf{N}$  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  $\xi$ , ranging from 0 to 1, as

$$\begin{aligned} N_1 &= 1 - \xi & N_2 &= \xi \\ N_3 &= 1 - 3\xi^2 + 2\xi^3 & N_4 &= 3\xi^2 - 2\xi^3 & N_5 &= L(\xi - 2\xi^2 + \xi^3) & N_6 &= L(-\xi^2 + \xi^3) \\ M_1 &= 1 - \xi & M_2 &= \xi \\ M_3 &= -\frac{6}{L}(\xi - \xi^2) & M_4 &= \frac{6}{L}(\xi - \xi^2) & M_5 &= 1 - 4\xi + 3\xi^2 & M_6 &= -2\xi + 3\xi^2 \end{aligned}$$

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

$$\begin{aligned} \Phi_2 &= \frac{12EI_{zz}}{G\kappa_y AL^2} = \frac{24(1+\nu)I_{zz}}{\kappa_y AL^2} \\ \Phi_3 &= \frac{12EI_{yy}}{G\kappa_z AL^2} = \frac{24(1+\nu)I_{yy}}{\kappa_z AL^2} \end{aligned}$$

which represent the ratios between bending and shear stiffness in the two principal directions. The shape functions are then modified so that

$$\begin{aligned}
N_3^i &= \frac{(N_3 + \Phi_i \hat{N}_3)}{1 + \Phi_i} & N_4^i &= \frac{(N_4 + \Phi_i \hat{N}_4)}{1 + \Phi_i} \\
N_5^i &= \frac{(N_5 + \Phi_i \hat{N}_5)}{1 + \Phi_i} & N_6^i &= \frac{(N_6 + \Phi_i \hat{N}_6)}{1 + \Phi_i} \\
M_5^i &= \frac{(M_5 + \Phi_i \hat{M}_5)}{1 + \Phi_i} & M_6^i &= \frac{(M_6 + \Phi_i \hat{M}_6)}{1 + \Phi_i}
\end{aligned}$$

where

$$\begin{aligned}
\hat{N}_3 &= 1 - \xi & \hat{N}_4 &= \xi \\
\hat{N}_5 &= \frac{L}{2}(\xi - \xi^2) & \hat{N}_6 &= -\frac{L}{2}(\xi - \xi^2) \\
\hat{M}_5 &= 1 - \xi & \hat{M}_6 &= \xi
\end{aligned}$$

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,  $\theta$ . 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  $\Theta_M$  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

$$[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3] = \mathbf{R}_r [\mathbf{r}_1^0 \ \mathbf{r}_2^0 \ \mathbf{r}_3^0]$$

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  $\xi$  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

$$\varepsilon = z_1 \frac{\partial \theta_{y1}}{\partial s} - y_1 \frac{\partial \theta_{z1}}{\partial s} + \frac{\partial u_{\text{axi}}}{\partial s}$$

where  $s$  is the local coordinate along the beam axis, and  $u_{\text{axi}}$  is the axial displacement. The coordinates from the beam centerline in the local transversal directions are denoted  $z_1$  and  $y_1$  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  $\varepsilon$  consists of thermal ( $\varepsilon_{\text{th}}$ ), hygroscopic ( $\varepsilon_{\text{hs}}$ ), initial ( $\varepsilon_i$ ), and elastic strains ( $\varepsilon_{\text{el}}$ )

$$\varepsilon = \varepsilon_{\text{el}} + \varepsilon_{\text{th}} + \varepsilon_{\text{hs}} + \varepsilon_i$$

### *Stress-Strain Relation*

---

The stress-strain relation for the axial deformation in the beam is described by

$$\sigma = E \varepsilon_{\text{el}} + \sigma_i$$

where  $E$  is Young's modulus, and  $\sigma_i$  is the initial stress.

The stress strain relation for the torsional and shear deformation is

$$\tau = G \gamma$$

where  $\tau$  is the shear stress,  $\gamma$  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  $\nu$ , 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

$$T = T_m + T_{gz}z_1 + T_{gy}y_1$$

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

$$\varepsilon_{th} = \alpha(T_m + T_{gz}z_1 + T_{gy}y_1 - T_{ref})$$

For the 2D beam, the term depending on  $z_1$  disappears.

### *Hygroscopic Swelling*

---

The moisture concentration is assumed to vary linearly across the beam's cross section. For the 3D beam it becomes

$$c_{mo} = c_m + c_{gz}z_1 + c_{gy}y_1$$

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

$$\varepsilon_{hs} = \beta_h(c_m + c_{gz}z_1 + c_{gy}y_1 - c_{mo,ref})$$

where  $\beta_h$  is the coefficient of hygroscopic swelling, and  $c_{mo,ref}$  is the strain-free reference moisture concentration.

For the 2D beam, the term depending on  $z_1$  disappears.

## Initial Load 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.

$$\begin{aligned}M_{iy1} &= \int_A \sigma_i z_1 dA \\M_{iz1} &= -\int_A \sigma_i y_1 dA \\N_i &= \int_A \sigma_i dA \\M_{ix1} &= \int_A (\tau_{ixz} y_1 - \tau_{iyz} z_1) dA\end{aligned}$$

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.

$$\begin{aligned}T_{iy1} &= \int_A \tau_{ixy} dA \\T_{iz1} &= \int_A \tau_{ixz} dA\end{aligned}$$

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

$$\varepsilon_i = z_1 \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - y_1 \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i + \left( \frac{\partial u_{axi}}{\partial s} \right)_i$$

In 2D the  $z_1$  dependent term disappears. As initial strain for the torsional degree of freedom, the derivative of the twist angle,

$$\left( \frac{\partial \theta_{x1}}{\partial s} \right)_i$$

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:

$$\delta W = \delta \int_V (-\varepsilon_{e1} \sigma + \mathbf{u}^T \mathbf{F}) dV = 0$$

The beam elements are formulated in terms of the stress resultants (normal force, bending moments and twisting moment).

The normal force is defined as

$$\begin{aligned} N &= \int_A \sigma dA = \int_A (E\varepsilon_{e1} + \sigma_i) dA = \\ & \int_A \left\{ E \left[ \left( z_1 \frac{\partial \theta_{y1}}{\partial s} - y_1 \frac{\partial \theta_{z1}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right) - \left[ z_1 \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - y_1 \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i + \left( \frac{\partial u_{axi}}{\partial s} \right)_i \right] - \right. \\ & \left. \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{mo,ref}) \right] + \sigma_i \right\} dA = \\ & \int_A \left\{ E \left[ \left( \frac{\partial u_{axi}}{\partial s} - \left( \frac{\partial u_{axi}}{\partial s} \right)_i \right) - \alpha (T_m - T_{ref}) - \beta_h (c_m - c_{mo,ref}) \right] \right\} dA + \int_A \sigma_i dA = \\ & EA \left[ \left( \frac{\partial u_{axi}}{\partial s} - \left( \frac{\partial u_{axi}}{\partial s} \right)_i \right) - \alpha (T_m - T_{ref}) - \beta_h (c_m - c_{mo,ref}) \right] + N_i \end{aligned}$$

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

$$\begin{aligned}
M_{y1} &= \int_A \sigma z_1 dA = \int_A z_1 (E \varepsilon_{el} + \sigma_i) dA = \\
&\int_A z_1 \left\{ E \left[ \left( z_1 \frac{\partial \theta_{y1}}{\partial s} - y_1 \frac{\partial \theta_{z1}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right) - \left[ z_1 \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - y_1 \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i + \left( \frac{\partial u_{axi}}{\partial s} \right)_i \right] - \right. \\
&\left. \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{mo,ref}) \right] + \sigma_i \right\} dA = \\
&\int_A z_1^2 \left\{ E \left[ \frac{\partial \theta_{y1}}{\partial s} - \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - \alpha T_{gz} - \beta_h c_{gz} \right] \right\} dA + \int_A \sigma_i z_1 dA = \\
&EI_{yy} \left[ \frac{\partial \theta_{y1}}{\partial s} - \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - \alpha T_{gz} - \beta_h c_{gz} \right] + M_{iy1}
\end{aligned}$$

$$\begin{aligned}
M_{z1} &= - \int_A \sigma y_1 dA = - \int_A y_1 (E \varepsilon_{el} + \sigma_i) dA = \\
&- \int_A y_1 \left\{ E \left[ \left( z_1 \frac{\partial \theta_{y1}}{\partial s} - y_1 \frac{\partial \theta_{z1}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right) - \left[ z_1 \left( \frac{\partial \theta_{y1}}{\partial s} \right)_i - y_1 \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i + \left( \frac{\partial u_{axi}}{\partial s} \right)_i \right] - \right. \\
&- \left. \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{mo,ref}) \right] + \sigma_i \right\} dA = \\
&- \int_A \left\{ y_1^2 \cdot \left[ E \left( - \frac{\partial \theta_{z1}}{\partial s} + \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i - \alpha T_{gy} - \beta_h c_{gy} \right) \right] \cdot dA \right\} - \int_A \sigma_i y_1 dA = \\
&EI_{zz} \left[ \frac{\partial \theta_{z1}}{\partial s} - \left( \frac{\partial \theta_{z1}}{\partial s} \right)_i + \alpha T_{gy} + \beta_h c_{gy} \right] + M_{iz1}
\end{aligned}$$

$M_{y1}$  is present only in 3D, and so is the torsional moment  $M_{x1}$  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

$$M_{x1} = GJ \left[ \frac{\partial \theta_{x1}}{\partial s} - \left( \frac{\partial \theta_{x1}}{\partial s} \right)_i \right] + M_{ix1}$$

Using the beam moment and normal force the expression for the virtual work becomes very compact:

$$\delta W = -\int_L \left[ M_{y1} \delta \left( \frac{\partial \theta_{y1}}{\partial s} \right) + M_{z1} \delta \left( \frac{\partial \theta_{z1}}{\partial s} \right) + N \delta \left( \frac{\partial u_{axi}}{\partial s} \right) + M_{x1} \delta \left( \frac{\partial \theta_{x1}}{\partial s} \right) \right] dx$$

For 2D, the first and fourth terms are omitted. For the case of Timoshenko beam, there is also a shear stress contribution added,

$$\delta W = -\int_L (T_{y1} \delta \gamma_{y1} + T_{z1} \delta \gamma_{z1}) dx$$

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 ( $f_{x1}, f_{y1}, f_{z1}$ ), which is defined as acting along the centerline, is then augmented by a twisting moment given by

$$m_{x1} = f_{y1} e_z - f_{z1} e_y$$

### Theory for Section Stiffness

---

The [Section Stiffness](#) node in the Beam interface allows modeling of beams with non-homogeneous cross sections by, instead of geometrical and material properties, directly entering values for the stiffness matrix  $\mathbf{S}$ .

For a 3D Euler-Bernoulli beam, the relationship between the section forces and the deformation of the beam is given by

$$\begin{bmatrix} N \\ M_{x1} \\ M_{y1} \\ M_{z1} \end{bmatrix} = \begin{bmatrix} N_i \\ M_{ix1} \\ M_{iy1} \\ M_{iz1} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \frac{\partial u_{axi}}{\partial s} - \epsilon_{ni} \\ \frac{\partial \theta_{x1}}{\partial s} - \theta_{six1} \\ \frac{\partial \theta_{y1}}{\partial s} - \theta_{siy1} \\ \frac{\partial \theta_{z1}}{\partial s} - \theta_{siz1} \end{bmatrix} \quad (8-1)$$

including initial normal force  $N_i$ ; moments  $M_{ixl}$ ,  $M_{iy1}$ , and  $M_{izl}$ ; normal strain  $\epsilon_i$ ; and curvatures  $\theta_{sixl}$ ,  $\theta_{siyl}$ , and  $\theta_{sizl}$ . Here,  $\mathbf{S}$  is a 4 by 4 symmetric matrix. For a homogeneous prismatic beam with isotropic material properties, it is defined as

$$\mathbf{S} = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 \\ 0 & 0 & EI_{yy} & 0 \\ 0 & 0 & 0 & EI_{zz} \end{bmatrix}$$

For a 2D Euler-Bernoulli beam, out-of-plane moments are zero, and [Equation 8-1](#) thus simplifies to

$$\begin{bmatrix} N \\ M_{zl} \end{bmatrix} = \begin{bmatrix} N_i \\ M_{izl} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \frac{\partial u_{axi}}{\partial s} - \epsilon_{ni} \\ \frac{\partial \theta_{zl}}{\partial s} - \theta_{sizl} \end{bmatrix}$$

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

$$\begin{bmatrix} N \\ T_{y1} \\ T_{zl} \\ M_{xl} \\ M_{y1} \\ M_{zl} \end{bmatrix} = \begin{bmatrix} N_i \\ T_{iy1} \\ T_{izl} \\ M_{ixl} \\ M_{iy1} \\ M_{izl} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \frac{\partial u_{axi}}{\partial s} - \epsilon_{ni} \\ \gamma_{y1} - \gamma_{iy1} \\ \gamma_{zl} - \gamma_{izl} \\ \frac{\partial \theta_{xl}}{\partial s} - \theta_{sixl} \\ \frac{\partial \theta_{y1}}{\partial s} - \theta_{siyl} \\ \frac{\partial \theta_{zl}}{\partial s} - \theta_{sizl} \end{bmatrix} \quad (8-2)$$

with  $\mathbf{S}$  being a 6 by 6 symmetric matrix. The equation now also includes the initial shear forces  $T_{iy1}$  and  $T_{izl}$  as well as the initial shear strains  $\gamma_{iy1}$  and  $\gamma_{izl}$ .

For a 2D Timoshenko beam, out-of-plane forces and moments are zero, and [Equation 8-2](#) thus simplifies to

$$\begin{bmatrix} N \\ T_{y1} \\ M_{z1} \end{bmatrix} = \begin{bmatrix} N_i \\ T_{iy1} \\ M_{iz1} \end{bmatrix} + \mathbf{S} \begin{bmatrix} \frac{\partial u_{axi}}{\partial s} - \epsilon_{ni} \\ \gamma_{y1} - \gamma_{iy1} \\ \frac{\partial \theta_{z1}}{\partial s} - \theta_{siz1} \end{bmatrix}$$

with  $\mathbf{S}$  now being a 3 by 3 symmetric matrix.

Using the section forces, the virtual work for the beam becomes

$$\begin{aligned} \delta W = - \int_L \left[ N \delta \left( \frac{\partial u_{axi}}{\partial s} \right) + T_{y1} \delta \gamma_{y1} + T_{z1} \delta \gamma_{z1} \right. \\ \left. + M_{x1} \delta \left( \frac{\partial \theta_{x1}}{\partial s} \right) + M_{y1} \delta \left( \frac{\partial \theta_{y1}}{\partial s} \right) + M_{z1} \delta \left( \frac{\partial \theta_{z1}}{\partial s} \right) \right] dx \end{aligned}$$

where only the relevant terms are kept, depending on the beam formulation and the spatial 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

$$\sigma_n = \frac{N}{A}$$

The normal stress from bending is computed in four user-selected points  $(y_{1k}, z_{1k})$  in the cross section as

$$\sigma_{bk} = \frac{M_{y1} z_{1k}}{I_{yy}} - \frac{M_{z1} y_{1k}}{I_{zz}}$$

In 2D, only two points, specified by their local  $y$ -coordinates are used.

The total normal stress in these points is then

$$\sigma_k = \sigma_{bk} + \sigma_n$$

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

$$\sigma_{b, \max} = \frac{d_o \sqrt{M_{y1}^2 + M_{z1}^2}}{2I_{zz}}$$

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

$$\tau_{t, \max} = \frac{|M_{x1}|}{W_t}$$

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.

$$T_{z1} = EI_{yy} \frac{\partial^2 \theta_{y1}}{\partial s^2}$$

$$T_{y1} = -EI_{zz} \frac{\partial^2 \theta_{z1}}{\partial s^2}$$

where  $T_{z1}$  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

$$\tau_{sz, \text{ave}} = \frac{T_{z1}}{A}$$

$$\tau_{sy, \text{ave}} = \frac{T_{y1}}{A} \tag{8-3}$$

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\begin{aligned}\tau_{sz, \max} &= \mu_z \tau_{sz, \text{ave}} \\ \tau_{sy, \max} &= \mu_y \tau_{sy, \text{ave}}\end{aligned}\quad (8-4)$$

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

$$\sigma_{\text{mises}} = \sqrt{\sigma_{\max}^2 + 3\tau_{xy, \max}^2 + 3\tau_{xz, \max}^2}$$

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.

### *Limited Displacement*

---

The **Limited** option that is available for prescribing displacements in the [Prescribed Displacement/Rotation](#) node is implemented as a weak inequality constraint. For each displacement component  $u_i$ , a gap distance is computed as

$$g_i = \min(-u_i + u_{0i, \max}, u_i - u_{0i, \min}) \quad (8-5)$$

where  $u_{0i, \max}$  and  $u_{0i, \min}$  are the maximum and minimum limits. Given [Equation 8-5](#), the weak inequality constraint is formulated as

$$0 = \int f_i \delta g_i dX \quad (8-6)$$

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).

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 8-6 is added as a weak contribution to the model to implement the constraint.

The augmented Lagrangian implementation of the inequality constraint is based on the following augmentation of Equation 8-6

$$\begin{aligned} 0 &= \int f_i \delta g_i dX \\ 0 &= -\frac{1}{k_p} \int (f_i - f_{lm,i}) \delta f_{lm,i} dX \end{aligned} \tag{8-7}$$

where  $f_{lm,i}$  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

$$f_i = \max(-k_p g_i + f_{lm,i}, 0)$$

and the constraint is implemented by adding the weak contributions defined in Equation 8-7 to the model.

### *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

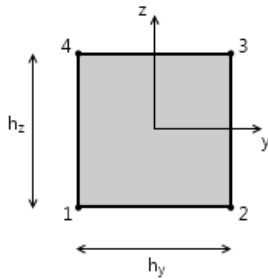


Figure 8-7: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-2: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$h_y h_z$	
$I_{zz}$	$\frac{h_y^3 h_z}{12}$	
$e_z$	0	
$\mu_y$	1.5	
$\kappa_y$	5/6	
$I_{yy}$	$\frac{h_z^3 h_y}{12}$	
$e_y$	0	
$\mu_z$	1.5	
$\kappa_z$	5/6	
$J$	$\frac{h_y^2 h_z^2 q}{3} \left( 1 - \sum_{n=1}^{\infty} \frac{192q}{\pi^5 (2n-1)^5} \tanh\left(\frac{(2n-1)\pi}{2q}\right) \right)$	$q = \min(h_y/h_z, h_z/h_y)$

TABLE 8-2: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$W_t$	$\frac{J}{\sqrt{h_y h_z q} \left( 1 - \sum_{n=1}^{\infty} \frac{8}{\pi^2 (2n-1)^2 \cosh\left(\frac{(2n-1)\pi}{2q}\right)} \right)}$	$q = \min(h_y/h_z, h_z/h_y)$
$p_1$	$\left( \frac{-h_y}{2}, \frac{-h_z}{2} \right)$	
$p_2$	$\left( \frac{h_y}{2}, \frac{-h_z}{2} \right)$	
$p_3$	$\left( \frac{h_y}{2}, \frac{h_z}{2} \right)$	
$p_4$	$\left( \frac{-h_y}{2}, \frac{h_z}{2} \right)$	
$r_e$	$\frac{h_y + h_z}{4}$	

**BOX SECTION**

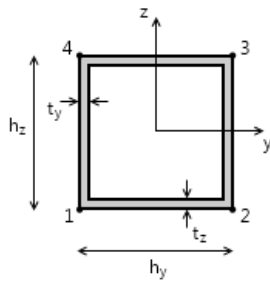


Figure 8-8: Geometry of a box shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-3: BOX SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$2(h_y t_z + h_z t_y) - 4t_y t_z$	
$I_{zz}$	$\frac{t_z h_y^3 + t_y^3 (h_z - 2t_z)}{6} + \frac{t_y (h_z - 2t_z) (h_y - t_y)^2}{2}$	
$e_z$	0	
$\mu_y$	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - 2t_z))A}{16t_z I_{zz}}$	
$\kappa_y$	$\frac{2h_y t_z}{A}$	
$I_{yy}$	$\frac{t_y h_z^3 + t_z^3 (h_y - 2t_y)}{6} + \frac{t_z (h_y - 2t_y) (h_z - t_z)^2}{2}$	
$e_y$	0	
$\mu_z$	$\frac{(h_z^2 h_y - (h_z - 2t_z)^2 (h_y - 2t_y))A}{16t_y I_{yy}}$	
$\kappa_z$	$\frac{2h_z t_y}{A}$	
$J$	$\frac{2(h_y - t_y)^2 (h_z - t_z)^2}{\frac{h_y - t_y}{t_z} + \frac{h_z - t_z}{t_y}}$	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 8-3: 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**

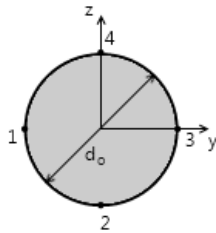


Figure 8-9: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

C

TABLE 8-4: CIRCULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$\frac{\pi d_0^2}{4}$	
$I_{zz}$	$\frac{\pi d_0^4}{64}$	
$e_z$	0	
$\mu_y$	$\frac{4}{3}$	
$\kappa_y$	0.9	
$I_{yy}$	$I_{zz}$	
$e_y$	0	

TABLE 8-4: CIRCULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$\mu_z$	$\mu_y$	
$\kappa_z$	$\kappa_y$	
$J$	$\frac{\pi d_o^4}{32}$	
$W_t$	$\frac{\pi d_o^3}{16}$	
$p_1$	$(-\frac{d_o}{2}, 0)$	
$p_2$	$(0, -\frac{d_o}{2})$	
$p_3$	$(\frac{d_o}{2}, 0)$	
$p_4$	$(0, \frac{d_o}{2})$	
$r_e$	$\frac{d_o}{2}$	

**PIPE SECTION**

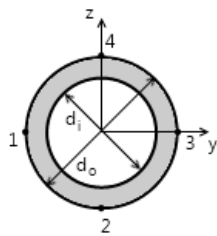


Figure 8-10: Geometry of a pipe cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-5: 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	
$\mu_y$	$\frac{(d_o^3 - d_i^3)A}{12(d_o - d_i)I_{zz}}$	
$\kappa_y$	$1 + \frac{d_i}{d_o}$	
$I_{yy}$	$I_{zz}$	
$e_y$	0	
$\mu_z$	$\mu_y$	
$\kappa_z$	$\kappa_y$	
$J$	$\frac{\pi(d_o^4 - d_i^4)}{32}$	
$W_t$	$\frac{\pi(d_o^4 - d_i^4)}{16d_o}$	
$p_1$	$\left(-\frac{d_o}{2}, 0\right)$	
$p_2$	$\left(0, \frac{-d_o}{2}\right)$	
$p_3$	$\left(\frac{d_o}{2}, 0\right)$	

TABLE 8-5: PIPE SECTION CONSTANTS

PROPERTY	FORMULA	REMARKS
$p_4$	$\left(0, \frac{d_o}{2}\right)$	
$r_e$	$\frac{d_o}{2}$	

**H-PROFILE SECTION**

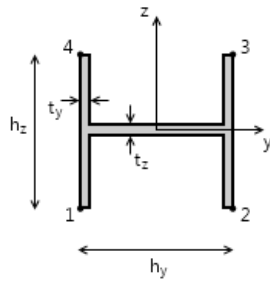


Figure 8-11: Geometry of an H-profile cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-6: 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	
$\mu_y$	$\frac{[4h_z t_y (h_y - t_y) + t_z (h_y - 2t_y)^2] A}{8t_z I_{zz}}$	
$\kappa_y$	$\frac{h_y t_z}{A}$	

TABLE 8-6: H-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$I_{yy}$	$\frac{2t_y h_z^3 + t_z^3 (h_y - 2t_y)}{12}$	
$e_y$	0	
$\mu_z$	$\frac{(h_z^2 - t_z^2)A}{8I_{yy}}$	
$\kappa_z$	$\frac{5h_z t_y}{3A}$	
$J$	$\frac{2t_y^3 h_z + t_z^3 (h_y - 2t_y)}{3}$	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

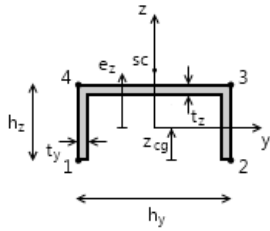


Figure 8-12: Geometry of a U-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-7: U-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$h_y t_z + 2(h_z - t_z) t_y$	
$I_{zz}$	$\frac{t_z h_y^3}{12} + 2t_y^3 (h_z - t_z) + \frac{t_y (h_z - t_z) (h_y - t_y)^2}{2}$	
$z_{CG}$	$\frac{h_y t_z (2h_z - t_z) + 2t_y (h_z - t_z)^2}{2A}$	
$e_z$	$h_z - \frac{t_z}{2} + \frac{t_y (2h_z - t_z)^2 (h_y - t_y)^2}{16I_{zz}} - z_{CG}$	
$\mu_y$	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - t_z)) A}{8t_z I_{zz}}$	
$\kappa_y$	$\frac{h_y t_z}{A}$	
$I_{yy}$	$\frac{8t_y h_z^3 + t_z^3 (h_y - 2t_y) + 3t_z (h_y - 2t_y) (2h_z - t_z)^2}{12}$ $- z_{CG}^2 A$	
$e_y$	0	

TABLE 8-7: U-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$\mu_z$	$\frac{z_{CG}^2 A}{2I_{yy}}$	
$\kappa_z$	$\frac{5h_z t_y}{3A}$	
$J$	$\frac{2t_y^3 h_z + t_z^3 (h_y - 2t_y)}{3}$	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

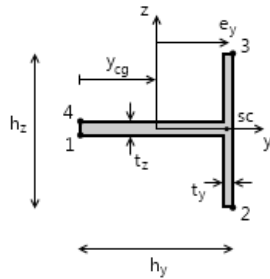


Figure 8-13: Geometry of a T-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-8: 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	
$\mu_y$	$\frac{y_{CG}^2 A}{2I_{zz}}$	
$\kappa_y$	$\frac{h_y t_z}{A}$	
$I_{yy}$	$\frac{t_y h_z^3 + t_z^3 (h_y - t_y)}{12}$	
$e_y$	$h_y - \frac{t_y}{2} - y_{CG}$	

TABLE 8-8: T-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$\mu_z$	$\frac{(h_z^2 - t_z^2)A}{8I_{yy}}$	
$\kappa_z$	$\frac{4h_z t_y}{3A}$	
$J$	$\frac{t_y^3 h_z + t_z^3 (h_y - t_y)}{3}$	Thin-walled approximation
$W_t$	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
$p_1$	$\left(-y_{CG}, \frac{-t_z}{2}\right)$	
$p_2$	$\left(h_y - y_{CG}, \frac{-h_z}{2}\right)$	
$p_3$	$\left(h_y - y_{CG}, \frac{h_z}{2}\right)$	
$p_3$	$\left(-y_{CG}, \frac{t_z}{2}\right)$	
$r_e$	$\frac{h_y + h_z}{4}$	

## C-PROFILE SECTION

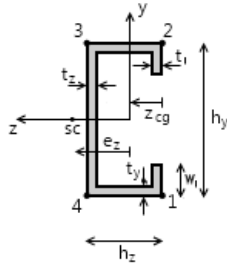


Figure 8-14: Geometry of a C-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-9: 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
$e_z$		See below the table
$\mu_y$	$\frac{[h_y^2 t_z + 4(h_z - t_z) t_y (h_y - t_y) + 4(w_1 - t_y) t_1 (h_y - t_y - w_1)] \frac{A}{8t_z I_{zz}}}{A}$	q
$\kappa_y$	$\frac{(h_y - 2t_y) t_z}{A}$	q

TABLE 8-9: C-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$I_{yy}$	$\frac{t_z^3 h_y + 8(h_z - t_z)^3 t_y + 8t_1^3 (w_1 - t_y)}{12} + \frac{3t_z h_y (2h_z - t_z)^2}{12} - z_{CG}^2 A$	
$e_y$	0	
$\mu_z$	$\frac{(t_y z_{CG}^2 + (w_1 - t_y)t_1(2z_{CG} - t_1))A}{2t_y I_{yy}}$	
$\kappa_z$	$\frac{5h_z t_y}{3A}$	
$J$	$\frac{t_z^3 h_y + 2t_y^3 (h_z - t_z) + 2t_1^3 (w_1 - t_y)}{3}$	Thin-walled approximation
$W_t$	$\frac{J}{\max(t_y, t_z, t_1)}$	Thin-walled approximation
$p_1$	$\left(-\frac{h_y}{2}, -z_{CG}\right)$	
$p_2$	$\left(\frac{h_y}{2}, -z_{CG}\right)$	
$p_3$	$\left(\frac{h_y}{2}, h_z - z_{CG}\right)$	
$p_4$	$\left(-\frac{h_y}{2}, h_z - z_{CG}\right)$	
$r_e$	$\frac{h_y + h_z}{4}$	

The expression for the shear center offset is valid only for the case when the thickness is constant, that is

$$t_y = t_z = t_1 = t$$

Then

$$e_z = h_z - \frac{t_z}{2} + \frac{N}{D} - z_{CG}$$

where

$$N = 2(h_z - t)(6(h_y - t)^2(h_z - t) + 6(h_y - t)^2(2w_1 - t) - 2(2w_1 - t)^3)$$

$$D = 4((h_y - t)^3 + 6(h_y - t)^2(h_z - t) + 3((h_y - t)^2(2w_1 - t)) + (2w_1 - t)^3) +$$

$$-12(h_y - t)(h_z - t)^2$$

### HAT SECTION

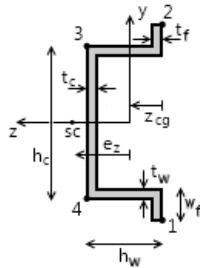


Figure 8-15: Geometry of a Hat cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-10: HAT SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$h_c t_c + 2(h_w - t_c)t_w + 2(w_f - t_w)t_f$	
$I_{zz}$	$\frac{t_c h_c^3 + 2t_w^3(h_w - t_c) + 2(w_f - t_w)t_f^3}{12} +$ $\frac{t_w(h_w - t_c)(h_c - t_w)^2 + t_f(w_f - t_w)(h_c + w_f)^2}{2}$	
$z_{CG}$	$\frac{h_c t_c(2h_w - t_c) + 2t_w(h_w - t_c)^2 + 2(w_f - t_w)t_f^2}{2A}$	

TABLE 8-10: HAT SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$e_z$		See below the table
$\mu_y$	$\left[ h_c^2 t_c + 4(h_w - t_c)t_w(h_c - t_w) + 4(w_f - t_w)t_f(h_c + w_f) \right] \frac{A}{8t_c I_{zz}}$	
$\kappa_y$	$\frac{(h_c - 2t_w)t_c}{A}$	
$I_{yy}$	$\frac{t_c^3 h_c + 8(h_w - t_c)^3 t_w + 8t_f^3 (w_f - t_w)}{12} + \frac{3t_c h_c (2h_w - t_c)^2}{12} - z_{CG}^2 A$	
$e_y$	0	
$\mu_z$	$\frac{(t_w z_{CG}^2 + (w_f - t_w)t_f(2z_{CG} - t_f))A}{2t_w I_{yy}}$	
$\kappa_z$	$\frac{5h_w t_w}{3A}$	
$J$	$\frac{t_c^3 h_c + 2t_w^3 (h_w - t_c) + 2t_f^3 (w_f - t_w)}{3}$	Thin-walled approximation
$W_t$	$\frac{J}{\max(t_y, t_z, t_f)}$	Thin-walled approximation
$p_1$	$\left( \frac{-(h_c + w_f - t_w)}{2}, -z_{CG} \right)$	
$p_2$	$\left( \frac{(h_c + w_f - t_w)}{2}, -z_{CG} \right)$	
$p_3$	$\left( \frac{h_y}{2}, h_z - z_{CG} \right)$	



TABLE 8-10: HAT 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

$$t_c = t_w = t_f = t$$

Then



$$e_z = h_w - \frac{t_z}{2} + \frac{N}{D} - z_{CG}$$

where

$$N = 2(h_w - t)(6(h_c - t)^2(h_w - t) + 6(h_c - t)^2(2w_f - t) - 2(2w_f - t)^3)$$

$$D = 4((h_c - t)^3 + 6(h_c - t)^2(h_w - t) + 3((h_c - t)^2(2w_f - t)) + (2w_f - t)^3) + 12(h_c - t)(h_w - t)^2$$

# 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`.

## 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.

## REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

## 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.

## 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 nodes. This means that it can, and often does, connect elements from different physics interfaces.

In the drop-down 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

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 ( $u, v, w$ ) in 3D.
- The rotation angle  $\theta$ , which has one component in 2D ( $th$ ) and three components in 3D ( $thx, thy, thz$ ).

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 through 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](#)
- [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
- Pinned
- Point Load
- Point Mass
- Point Mass Damping
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Prescribed Acceleration
- Rigid Connector
- Section Orientation
- Section Stiffness
- Shell Connection
- Solid Connection
- Symmetry
- Thermal Expansion (for Materials)

These nodes are described for the Solid Mechanics interface:

- Added Mass
- Center of Rotation Nodes
- Damping
- Fixed Constraint
- Free
- Gravity<sup>1</sup>
- Predeformation
- Rigid Domain
- Rotating Frame<sup>1</sup>
- Safety
- Spring Foundation
- Spring-Damper

<sup>1</sup> 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 adds an initial 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**  $\mathbf{u}$
- **Velocity field**  $\frac{d\mathbf{u}}{dt}$
- **Rotation field**  $\theta$
- **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

The default is **User defined**. Select **Common sections** to choose from predefined sections.

For **User defined** go to [Basic Section Properties](#) and [Stress Evaluation Properties](#) to continue defining the cross section.

For **Common sections** select a **Section type** — **Rectangle**, **Box**, **Circular**, **Pipe**, **H-profile**, **U-profile**, **T-profile**, **C-profile**, or **Hat**. Then go to the relevant section below to continue defining the section. Each **Section type** 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**  $t_f$

#### **BASIC 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	$A$	$\text{m}^2$
2D and 3D	Moment of inertia about local z-axis	$I_{zz}$	$\text{m}^4$
2D and 3D, Timoshenko beam	Shear correction factor along local y-axis	$\kappa_y$	l
3D only	Distance to shear center in local z direction	$e_z$	m
3D only	Moment of inertia about local y-axis	$I_{yy}$	$\text{m}^4$
3D only	Distance to shear center in local y direction	$e_y$	m
3D only	Torsional constant	$J$	$\text{m}^4$
3D only, Timoshenko beam	Shear correction factor along local z-axis	$\kappa_z$	l



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 **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 8-3](#) and [Equation 8-4](#).

*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	$\mu_y$	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	$\mu_z$	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	$\mu_y$	l
3D only	Torsional section modulus	$W_t$	$m^3$
3D only	Max shear stress factor in local z direction	$\mu_z$	l

**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** For enter a **Reference point defining local y direction**  $P$ .

The coordinate system is defined as follows:

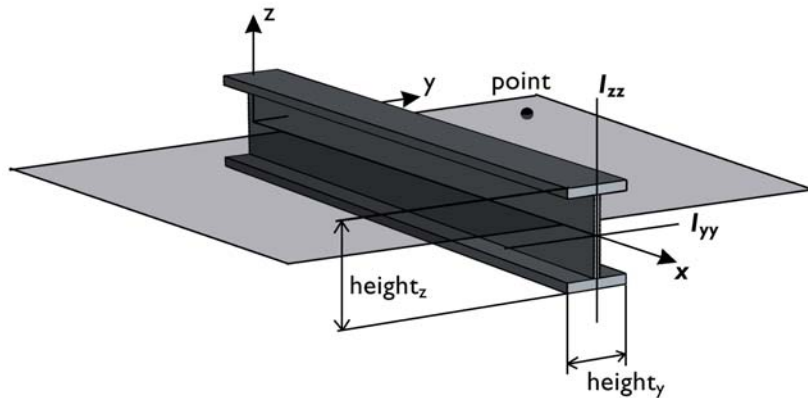
The local  $x$  direction is in the edge direction. The positive edge direction can be checked by vector plotting the local edge tangent direction. The coordinates of the reference point define the local  $xy$ -plane together with the beam axis. The local coordinate system ( $\mathbf{e}_{x1}$ ,  $\mathbf{e}_{y1}$ ,  $\mathbf{e}_{z1}$ ) is formed using the following algorithm:

$$\mathbf{v}_{z1} = \mathbf{e}_{x1} \times (\mathbf{p} - \mathbf{m})$$

$$\mathbf{e}_{z1} = \frac{\mathbf{v}_{z1}}{|\mathbf{v}_{z1}|}$$

$$\mathbf{e}_{y1} = \mathbf{e}_{z1} \times \mathbf{e}_{x1}$$

Here,  $\mathbf{p}$  is the reference point, and  $\mathbf{m}$  is the midpoint of the beam element. The definition of the local coordinate system is illustrated in [Figure 8-16](#).



*Figure 8-16: 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  $\phi$** . 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 ( $\mathbf{e}_{x1}$ ,  $\mathbf{e}_{y1}$ ,  $\mathbf{e}_{z1}$ ) is formed using the following algorithm:

$$\mathbf{v}_{z1} = \mathbf{e}_{x1} \times \mathbf{V}$$

$$\mathbf{e}_{z1} = \frac{\mathbf{v}_{z1}}{|\mathbf{v}_{z1}|}$$

$$\mathbf{e}_{y1} = \mathbf{e}_{z1} \times \mathbf{e}_{x1}$$

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*

**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](#)
- [Safety](#)
- [Damping](#)

## 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  $\nu$ . 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  $\lambda$  and  $\mu$ .
- **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**  $\rho$  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**

## 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)  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

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**  $T_{\text{ref}}$  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**,  $\alpha$ , list, select **From material** to use the coefficient of thermal expansion from the material, or **User defined** to enter a value or expression for  $\alpha$ .

### THERMAL BENDING

Enter the **Temperature gradient in local y direction**  $T_{\text{gy}}$  (in 2D and 3D) and in the **Temperature gradient in local z direction**  $T_{\text{gz}}$  (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  $\beta_h$  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**  $\beta_h$  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  $\beta_h$ . The default value for the **User defined** case is  $1.5e-4 \text{ m}^3/\text{kg}$ .

### 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**  $M_{iz}$  and for 3D models,  $M_{iy}$ .
- *For 3D models:* **Initial torsional moment**  $M_{ix}$ .
- *For Timoshenko beam:* **Initial shear force**  $T_{iy}$  and for 3D models,  $T_{iz}$ .

### INITIAL STRAIN

Specify the initial strain as the:

- **Initial axial strain**  $e_{ni}$ .
- **Initial curvature**  $\theta_{siz}$  and for 3D models, and  $\theta_{siy}$ .
- *For 3D models:* **Initial twist**  $\theta_{six}$ .
- *For Timoshenko beam:* **Initial shear strain**  $\gamma_{siy}$  and for 3D models,  $\gamma_{siz}$ .



For details about initial stresses and strains, see [Inelastic Strain Contributions](#) and [Initial Stresses and Strains](#).

---

### 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**  $N_{\text{ext}}$ .
- **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 spatial 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  $M_{x1}$ , and bending moments  $M_{y1}$  and  $M_{z1}$ . The rows, in order, correspond to the axial strain  $\epsilon_n$ , axial twist  $\kappa_{x1}$ , and bending curvatures  $\kappa_{y1}$  and  $\kappa_{z1}$ .
- 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  $M_{z1}$ . The rows, in order, correspond to the axial strain  $\epsilon_n$ , and curvature  $\kappa_{z1}$ .
- For a 3D Timoshenko beam, enter the 6 by 6 matrix. The columns, in order, correspond to the axial force  $N$ , shear forces  $T_{y1}$  and  $T_{z1}$ , torsional moment  $M_{x1}$ , and bending moments  $M_{y1}$  and  $M_{z1}$ . The rows, in order, correspond to the axial strain  $\epsilon_n$ , shear strains  $\gamma_{y1}$  and  $\gamma_{z1}$ , axial twist  $\kappa_{x1}$ , and bending curvatures  $\kappa_{y1}$  and  $\kappa_{z1}$ .
- For a 2D Timoshenko beam, enter the 3 by 3 matrix. The columns, in order, correspond to the axial force  $N$ , shear force  $T_{y1}$ , and bending moment  $M_{z1}$ . The rows, in order, correspond to the axial strain  $\epsilon_n$ , shear strain  $\gamma_{y1}$ , and bending curvature  $\kappa_{z1}$ .

In 3D, enter also 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 non-zero, 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**,  $\mathbf{I}_m$ , of the beam in order to define the inertial forces. In 2D, only the **Mass moment of inertia per unit length, zz component**,  $I_{m,zz}$ , is needed.

In addition, specify the **First moment of mass per unit length, local y component**,  $m_{1y}$ , and the **First moment of mass per unit length, local z component**,  $m_{1z}$ , 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](#)).

## 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  $M_{y1}$ , and the fourth to the bending moment around the local  $z$ -axis  $M_{z1}$ . 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  $M_{z1}$  are required.

In addition, in 3D enter a **Twisting moment factor**,  $S_T$ , a **Shear force factor, local y direction**,  $S_{S_y}$ , and a **Shear force factor, local z direction**,  $S_{S_z}$ . 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 to set maximum and minimum limits for the displacements, so that for example a one-sided support can be modeled.

- If a prescribed displacement or rotation 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  $\text{Inf}$  and  $-\text{Inf}$ , 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  
 $100 * \text{beam.Eequ} * \text{beam.area} * \text{beam.re}^2 / \text{beam.<tag>.charLen}^4$
- For edges, the default expression is  
 $100 * \text{beam.Eequ} * \text{beam.area} * \text{beam.re}^2 / \text{beam.<tag>.charLen}^3$

In these expressions, `beam` is the tag of the Beam interface and `<tag>` is the tag of the Prescribed Displacement/Rotation node. The expressions are given as an estimation to the bending stiffness of the beam, where `beam.Eequ` is the equivalent Young's modulus, `beam.re` is the equivalent radius of gyration, and `beam.area` is the cross-section area. The variable `beam.<tag>.charLen` is by default equal to the length of the mesh element. To improve the estimate of the penalty factor, replace `beam.<tag>.charLen` with the free length of the beam. The penalty factor can also be tuned by changing the multiplier at the beginning of each expression.



---

For details about setting 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** check box and enter a value or expression for  $\theta_{0z}$ .




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  $\theta$ . 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** check boxes to remove the constraint for the corresponding rotation component. If unchecked, the rotations are constrained to either the input value or to the default zero rotation. The status of the check boxes 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,  $\theta$  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 box.



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** check boxes. 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** check box 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** check boxes 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** check boxes. 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** check box 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** check boxes 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 box.



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** 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  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  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**  $\alpha$ . 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  $\alpha$ .

- A value or expression for the **Volume reference temperature**  $T_{\text{ref}}$  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 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**

**Pipe Mechanics>Fixed>Thermal Expansion**

**Pipe Mechanics>Prescribed Displacement>Thermal Expansion**

### *Ribbon*

Physics tab with **Fixed** or **Prescribed Displacement** 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 box.



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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry Constraints</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## 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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry Constraints</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## 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 unit length** (the default), **Force per unit volume**, or **Total force**. Enter values or expressions for the components.

TABLE 8-11:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit length	$F_L$	N/m	edges	3D (x, y, z)
			boundaries	2D (x, y)
Force per unit volume	$F_V$	N/m <sup>3</sup>	edges	3D (x, y, z)
			boundaries	2D (x, y)
Total force	$F_{tot}$	N	edges	3D (x, y, z)
			boundaries	2D (x, y)



- When **Force per unit 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 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 edge load  $M_L$  (3D) or  $M_{Lz}$  (2D).



- 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.

#### **POINT LOAD**

Enter values or expressions for the components (x, y, z for 3D models and x, y for 2D models) of the point load  $\mathbf{F}_P$ .

#### **POINT MOMENT**

Enter values or expressions for the components (x, y, z for 3D models and z for 2D models) of the point moment  $\mathbf{M}_P$  (3D) or  $M_{1z}$  (2D).



- The  $\mathbf{F}_P$  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.
- 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 Mass*

---

Use the **Point Mass** node to model a discrete mass or mass moment of inertia which is concentrated at a point.

The [Point Mass Damping](#) subnode can be added to specify a mass-proportional damping.

### 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.

### FRAME ACCELERATION FORCES

Click to select the **Exclude contribution** check box to switch off the loads that can be caused by the point mass when the frame is accelerated when using a [Gravity](#) or [Rotating Frame](#) feature. The setting will also determine whether the node will contribute when [Computing Mass Properties](#).

### 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**

## *Point Mass Damping*

---

Use the **Point Mass Damping** subnode to add damping to a [Point Mass](#) parent node.

### **POINT MASS DAMPING**

Enter a **Mass damping parameter**  $\alpha_{dM}$ . This is the mass proportional term of a Rayleigh damping.

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Beam>Point Mass>Point Mass Damping**

**Pipe Mechanics>Point Mass>Point Mass Damping**

#### *Ribbon*

Physics tab with **Point Mass** node selected in the model tree:

**Attributes>Point Mass Damping**

## *Solid Connection*

---



The **Solid Connection** node is obsolete, and has been superseded by the [Solid-Beam Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Solid Connection** node will be removed in future versions.

Use a **Solid Connection** node for connecting beams to solids in 2D models. To complete the connection, a corresponding **Beam Connection** node must also be added in the Solid Mechanics interface. The **Solid Connection** node can be added to either points or boundaries.

When using it for a point, the connection is intended for a transition from a beam to a solid, so that beam theory is assumed to be valid on both sides of the connection.

When using it for a boundary, the beam acts as a cladding or stiffener on the solid.

## SOLID CONNECTION (POINTS ONLY)

Select a **Connection type** — **Softened** or **Simplified**. When using **Softened** (the default), three extra degrees of freedom are added to each selected point. This allows for a more accurate description of the transition, but the model can in some cases become underconstrained if the mesh on the solid is very coarse. The **Simplified** version of the connection just adds constraints to the boundary of the solid, and will in general cause local disturbances of the stress field.

When using the **Solid Connection** node for a boundary, the connection is intended for placing a beam as a stiffener or cladding along the boundary of a solid. No other settings than the boundary selection are required.



- 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](#)

## LOCATION IN USER INTERFACE

*Context Menus*

**Beam>Connections>Solid Connection**

*Ribbon*

Physics tab with **Shell** selected:

**Boundaries>Connections>Solid Connection**

**Points>Connections>Solid Connection**

*Shell Connection*



The **Shell Connection** node is obsolete, and has been superseded by the [Shell-Beam Connection](#) multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Shell Connection** node will be removed in future versions.

A beam can be connected to a shell by adding a **Shell Connection** node in the Beam interface and a [Beam Connection](#) node in the Shell interface. The connection can be between:

- two edges, one in each physics interface,
- a point in the Beam interface and a boundary in the Shell interface, or
- a point in the Beam interface and an edge in the Shell interface.

There are no settings for the **Shell Connection** node except the selections of points or edges.



- For more information about coupling different element types, see [Coupling Techniques](#).
  - For details about the formulation of this coupling, see [Connection Between Shells and Beams](#)
- 

## LOCATION IN USER INTERFACE

### *Context Menus*

**Beam>Connections>Shell Connection**

### *Ribbon*

Physics tab with **Shell** selected:

**Edges>Connections>Shell Connection**

**Points>Connections>Shell Connection**

### *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.

The feature is similar to the rigid connectors in the Solid Mechanics and Shell interfaces. Rigid connectors from Beam, Pipe Mechanics, Shell, and Solid Mechanics 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 any additional coordinate systems that the model includes. 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.

## 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** check box to add an optional offset vector to the definition of the center of rotation. Enter values for the offset vector  $\mathbf{X}_{\text{offset}}$ .

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 at the center of rotation for each space direction, select one or several of the available check boxes 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**  $u_{0x}$
- **Prescribed in y direction**  $u_{0y}$
- *For 3D components:* **Prescribed in z direction**  $u_{0z}$

### **PRESCRIBED ROTATION AT CENTER OF ROTATION**

Specify the rotation at the center of rotation. Select from the **By** list: **Free** (the default), **Constrained rotation**, or **Prescribed rotation at center of rotation**.



For 2D components, the **Constrained rotation** and **Prescribed rotation at center of 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 check boxes 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 at Center of Rotation

- For **Prescribed rotation at center of rotation** enter an **Angle of rotation**  $\phi_0$ . For 3D components also enter an **Axis of rotation**  $\Omega$  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 Theory](#)
- [Harmonic Perturbation](#)
- [Load Cases](#)

### 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.

### 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.



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 Theory](#)

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Beam>Line Constraints>Rigid Connector** (3D: Edge, 2D: Boundary)

**Beam>Connections>Rigid Connector** (Point)

### *Ribbon*

Physics tab with **Beam** 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 **Attachment** node is available with either the Multibody Dynamics Module or the Rotordynamics Module. The use of attachments is discussed in the [Attachments](#) section in the *Multibody Dynamics Module User's Guide*.

## 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 box.

## 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.



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 ( $\phi$ ), 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 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  $\mathbf{M}_{\text{ph}}$  for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

## LOCATION IN USER INTERFACE

*Context Menus*

**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**



# 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, moments of inertia. In COMSOL Multiphysics there are two physics interfaces for analyzing beam structures, one in 2D and one in 3D.

### **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 **Common sections** option in 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 does, 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 have been used.

### STRESS COMPUTATION

Since the Beam interface is only aware of the cross-section properties and not of the true shape of the cross section, the stresses computed there can only be approximate. The approximation used by COMSOL is conservative in the sense that all stresses 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 exceed allowed values.

### *Using the Beam Cross Section Interface*

---

The Beam Cross Section interface can either be used separately or as a separate model node and geometry in the same model file where the actual beam problem is solved.



In most cases, it is easiest to compute the beam cross-section properties in a separate model.

---

There are some cases when using one or more Beam Cross Section interfaces and a Beam interface together in the same model file can be advantageous. For example, when changes in the cross sections can be anticipated. There are however a number of things to pay attention to in this case:

- Usually the determination of the cross-section data is more computationally expensive than the actual analyses of the beam structure. Then it is best to use either separate studies or study steps for the two tasks.
- When 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 click to clear the **Store in output** check box for the beam cross section degrees of freedom in order to save space.
- When referencing the beam cross-section properties in the input fields of the Beam interface, you must use a fully qualified variable name, like `mod2.bcs3.A` for the area.
- If a Beam interface is added after 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*

Since the Beam Cross Section interface is active 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.

In order to avoid confusion, the cross-section properties are described in local  $x_1$ - $x_2$  coordinates (see [Figure 9-1](#)). 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**

### COMPUTING THE CROSS-SECTION DATA

In a 2D model, the geometry of the cross section is drawn. If the section is simply connected (that is, has no internal holes), then usually nothing else needs to be done before running the analysis.

The default mesh density is tuned for thin-walled sections. For solid sections an unnecessarily large model is obtained when using the default mesh.

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 computed cross-section data is stored in the variables listed in [Table 9-1](#):

TABLE 9-1: VARIABLES CONTAINING CROSS-SECTION DATA

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
bcs.A	Area	Area (A)	<a href="#">Area</a>
bcs.CGx	Center of gravity, x coordinate	Implicit, since it affects the positioning of the beam centerline.	<a href="#">Center of Gravity</a>
bcs.CGy	Center of gravity, y coordinate	As above	

TABLE 9-1: VARIABLES CONTAINING CROSS-SECTION DATA

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
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 ( $I_{zz}/I_{yy}$ )	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 ( $\phi$ )	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	
bcs.J	Torsional constant	Torsional constant ( $J$ )	Torsion
bcs.Wt	Torsional section modulus	Torsional section modulus ( $W_t$ )	

TABLE 9-1: VARIABLES CONTAINING CROSS-SECTION DATA

VARIABLE	DESCRIPTION	INPUT TO BEAM INTERFACE	LINK TO THEORY SECTION
bc.s.Cw	Warping constant	-	<a href="#">Warping</a>
bc.s.Kw	Warping section modulus	-	
bc.s.sw	Nonuniform torsion parameter	-	

### COMPUTING DETAILED STRESSES

If you have 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 9-2](#).



After changing data in this section, you do not need to compute the study for this physics interface again once it 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 9-2: VARIABLES CONTAINING STRESS DISTRIBUTIONS

VARIABLE	DESCRIPTION AND	LINK TO THEORY SECTION
bc.s.sN	Stress from axial force	<a href="#">Axial Stress</a>
bc.s.sM1	Bending stress from moment around x1 axis	<a href="#">Bending Axial Stresses</a>
bc.s.sM2	Bending stress from moment around x2 axis	

TABLE 9-2: VARIABLES CONTAINING STRESS DISTRIBUTIONS

VARIABLE	DESCRIPTION AND	LINK TO THEORY SECTION
bcS.tT1x	Shear stress from force in x1 direction, x component	<a href="#">Bending Shear Stresses</a>
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	<a href="#">Torsional Shear Stresses</a>
bcS.tMty	Shear stress from twisting moment, y component	
bcS.resMt	Shear stress from twisting moment, resultant	
bcS.mises	von Mises equivalent stress	<a href="#">Equivalent Stress</a>
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`.

# 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:

$$x_{CG} = \frac{1}{A} \int_A x dA$$

$$y_{CG} = \frac{1}{A} \int_A y dA$$

### **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  $\alpha$ . 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  $-\pi < \alpha < \pi$ .

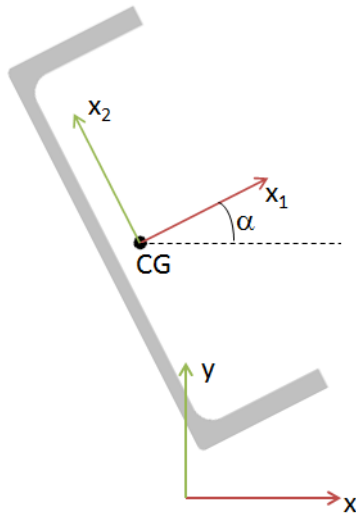


Figure 9-1: Local coordinate system and rotation angle.

### 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  $\kappa_1$  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 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:

$$\begin{aligned} 0 &= \int_A (-\tau_{1z}(e_2 - x_2) + \tau_{2z}(e_1 - x_1)) dA = \\ & \int_A (\tau_{1z}x_2 - \tau_{2z}x_1) dA + e_1 \int_A \tau_{2z} dA - e_2 \int_A \tau_{1z} dA \end{aligned}$$

Since there are two separate solutions for the shear stresses, it is possible to split the determination of the two shear center coordinates into:

$$\begin{aligned} e_1 &= - \int_A (\tau_{1z}^{(2)}x_2 - \tau_{2z}^{(2)}x_1) dA \\ e_2 &= \int_A (\tau_{1z}^{(1)}x_2 - \tau_{2z}^{(1)}x_1) dA \end{aligned}$$

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:

$$\begin{aligned}\sigma_1 &= 0 \\ \sigma_2 &= 0 \\ \sigma_z &= \frac{M_1 x_2}{I_1} - \frac{M_2 x_1}{I_2} \\ \tau_{12} &= 0\end{aligned}\tag{9-1}$$

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,  $\psi$ , 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:

$$\Delta\Psi = -\left(\frac{T_2x_2}{I_1} + \frac{T_1x_1}{I_2}\right) \quad (9-2)$$

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 9-1](#), the only two nontrivial equations are:

$$\begin{aligned} \Delta\tau_{1z} + \frac{1}{1+\nu} \frac{T_1}{I_2} &= 0 \\ \Delta\tau_{2z} + \frac{1}{1+\nu} \frac{T_2}{I_1} &= 0 \end{aligned}$$

Inserting the assumed stress components gives:

$$\begin{aligned} \Delta\Psi_{,1} + \frac{\nu T_1}{(1+\nu)I_2} + \frac{1}{1+\nu} \frac{T_1}{I_2} &= \Delta\Psi_{,1} + \frac{T_1}{I_2} = 0 \\ \Delta\Psi_{,2} + \frac{\nu T_2}{(1+\nu)I_1} + \frac{1}{1+\nu} \frac{T_2}{I_1} &= \Delta\Psi_{,2} + \frac{T_2}{I_1} = 0 \end{aligned}$$

Integration of the first equation with respect to  $x_1$  and the second equation with respect to  $x_2$  gives:

$$\begin{aligned} \Delta\Psi + \frac{T_1x_1}{I_2} + f_2(x_2) &= 0 \\ \Delta\Psi + \frac{T_2x_2}{I_1} + f_1(x_1) &= 0 \end{aligned}$$

Combining these two equations results in:

$$\Delta\Psi = -\left(\frac{T_2x_2}{I_1} + \frac{T_1x_1}{I_2}\right)$$

This is the same equation as [Equation 9-2](#). 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  $\psi$ , a term containing the differential equation itself is added. This is a zero contribution, but it makes further simplifications possible.

$$\begin{aligned}
\int_A \psi_{,1} dA &= \int_A \left( \psi_{,1} + x_1 \left( \Delta \psi + \frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2} \right) \right) dA = \\
\int_A (\psi_{,1} + x_1 \Delta \psi) dA + T_1 &= \int_A \left( \frac{\partial}{\partial x_1} (x_1 \psi_{,1}) + \frac{\partial}{\partial x_2} (x_1 \psi_{,2}) \right) dA + T_1 = \\
\oint_{\Gamma} ((x_1 \psi_{,1}) n_1 + (x_1 \psi_{,2}) n_2) ds + T_1 &= \oint_{\Gamma} x_1 (\nabla \psi \cdot \mathbf{n}) ds + T_1 = \\
\oint_{\Gamma} x_1 \left( - \left( \frac{\nu T_1 x_2^2}{2(1+\nu) I_2} n_1 + \frac{\nu T_2 x_1^2}{2(1+\nu) I_1} n_2 \right) \right) ds + T_1 &= \\
\int_A \left( - \frac{\nu T_1 x_2^2}{2(1+\nu) I_2} \right) dA + T_1 &= \frac{-\nu T_1 I_1}{2(1+\nu) I_2} + T_1
\end{aligned}$$

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:

$$\begin{aligned}\varepsilon_1 &= \frac{\partial u_1}{\partial x_1} = -\nu\varepsilon_z \\ \varepsilon_2 &= \frac{\partial u_2}{\partial x_2} = -\nu\varepsilon_z \\ \varepsilon_z &= \frac{\partial u_z}{\partial z} = \frac{M_1 x_2}{EI_1} - \frac{M_2 x_1}{EI_2} \\ \gamma_{12} &= \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} = 0 \\ \gamma_{1z} &= \frac{\partial u_1}{\partial z} + \frac{\partial u_z}{\partial x_1} = \frac{1}{G} \left( \Psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2} \right) \\ \gamma_{2z} &= \frac{\partial u_2}{\partial z} + \frac{\partial u_z}{\partial x_2} = \frac{1}{G} \left( \Psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1} \right)\end{aligned}$$

Integration of the direct strains gives:

$$\begin{aligned}u_1 &= -\frac{\nu M_1 x_1 x_2}{EI_1} + \frac{\nu M_2 x_1^2}{2EI_2} + g_1(x_2, z) \\ u_2 &= -\frac{\nu M_1 x_2^2}{2EI_1} + \frac{\nu M_2 x_1 x_2}{EI_2} + g_2(x_1, z)\end{aligned}$$

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  $z$ .

$$\begin{aligned}\oint_{\Gamma} du_1 &= \oint_{\Gamma} \left( \frac{\partial u_1}{\partial x_1} dx_1 + \frac{\partial u_1}{\partial x_2} dx_2 \right) = \\ \oint_{\Gamma} \left( \left( -\left( \frac{\nu M_1 x_2}{EI_1} - \frac{\nu M_2 x_1}{EI_2} \right) \right) dx_1 + \left( -\frac{\nu M_1 x_1}{EI_1} + \frac{\partial g_1}{\partial x_2} \right) dx_2 \right) &= \\ \int_A \left( \frac{\nu M_1 x_1 x_2}{EI_1} \right) dA - \int_A \left( \frac{\nu M_1 x_1 x_2}{EI_1} \right) dA &= 0\end{aligned}$$

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 9-3](#):

$$\begin{aligned}
 \oint_{\Gamma} du_z &= \oint_{\Gamma} \left( \frac{\partial u_z}{\partial x_1} dx_1 + \frac{\partial u_z}{\partial x_2} dx_2 \right) = \oint_{\Gamma} \left( \left( \gamma_{1z} - \frac{\partial u_1}{\partial z} \right) dx_1 + \left( \gamma_{2z} - \frac{\partial u_2}{\partial z} \right) dx_2 \right) = \\
 &\oint_{\Gamma} \left( \frac{1}{G} \left( \psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2} \right) - \left( \frac{\nu T_1 x_1 x_2}{EI_1} + \frac{\nu T_2 x_1^2}{2EI_2} \right) \right) dx_1 + \\
 &\oint_{\Gamma} \left( \frac{1}{G} \left( \psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1} \right) - \left( \frac{\nu T_1 x_2^2}{2EI_1} + \frac{\nu T_2 x_1 x_2}{EI_2} \right) \right) dx_2 = \tag{9-3} \\
 &- \int_A \left( \frac{1}{G} \left( \psi_{,12} + \frac{\nu T_1 x_2}{(1+\nu)I_2} \right) - \frac{\nu T_1 x_1}{EI_1} \right) dA + \\
 &\int_A \left( \frac{1}{G} \left( \psi_{,21} + \frac{\nu T_2 x_1}{(1+\nu)I_1} \right) - \frac{\nu T_2 x_2}{EI_2} \right) dA = 0
 \end{aligned}$$

In the last step of [Equation 9-3](#) 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:

$$\begin{aligned}
 \tau_{1z}^{(1)} &= \psi_{,1} + \frac{\nu x_2^2}{2(1+\nu)I_2} \\
 \tau_{2z}^{(1)} &= \psi_{,2}
 \end{aligned}$$

The corresponding problem for the  $x_2$  direction is:



$$\Delta \psi^{(2)} = -\frac{x_2}{I_1}$$

$$\nabla \psi^{(2)} \cdot \mathbf{n} = \frac{-vx_1^2}{2(1+\nu)I_1} n_1$$

$$\tau_{1z}^{(2)} = \psi_{,1}^{(2)}$$

$$\tau_{2z}^{(2)} = \psi_{,2}^{(2)} + \frac{vx_1^2}{2(1+\nu)I_1}$$

### 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 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  $\phi$  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 external 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  $\Gamma_i$ . This is trivially fulfilled for the in-plane displacements, but the out-of-plane displacement,  $w$ , generates the necessary equations to determine  $H_i$ :

$$\begin{aligned} 0 &= \oint_{\Gamma_i} dw = \oint_{\Gamma_i} \left[ \frac{\partial w}{\partial x} dx + \frac{\partial w}{\partial y} dy \right] = \oint_{\Gamma_i} \left[ \left( \gamma_{xz} - \left( \frac{\partial w}{\partial z} \right) \right) dx + \left( \gamma_{yz} - \left( \frac{\partial w}{\partial z} \right) \right) dy \right] \\ &= \oint_{\Gamma_i} \left[ \left( \frac{\partial \phi}{\partial y} + y \right) dx + \left( -\frac{\partial \phi}{\partial x} - x \right) dy \right] = \oint_{\Gamma_i} \left[ \left( \frac{\partial \phi}{\partial y} + y \right) t_x - \left( \frac{\partial \phi}{\partial x} + x \right) t_y \right] ds \end{aligned}$$

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:

$$\begin{aligned} u &= -yz \\ v &= xz \end{aligned}$$

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.

Since the gradient of  $\phi$  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 St 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 the Laplace 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 since 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,  $\phi_{x1}$  as input:

$$u(x,y) = \phi_{x1}\omega(x,y)$$

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:

$$\frac{\pi^2 E C_w}{G J L^2} = \frac{2(1+\nu)\pi^2 C_w}{J L^2}$$

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:

$$\tau_{T1} = \sqrt{\tau_{T1x}^2 + \tau_{T1y}^2} \quad \text{and} \quad \tau_{T2} = \sqrt{\tau_{T2x}^2 + \tau_{T2y}^2}$$

### TORSIONAL SHEAR STRESSES

The components of the shear stress caused by Saint-Venant torsion are:

$$\tau_{Mtx} = \frac{M_t \phi_{,y}}{J}$$

$$\tau_{Mty} = -\frac{M_t \phi_{,x}}{J}$$

where  $M_t$  is the twisting moment. The resultant is computed as:



$$\tau_{Mt} = \sqrt{\tau_{Mtx}^2 + \tau_{Mty}^2}$$

### EQUIVALENT STRESS

The von Mises equivalent stress is computed from the stress components defined above using the expression:

$$\sigma_{vM} = \sqrt{(\sigma_N + \sigma_{M1} + \sigma_{M2})^2 + 3(\tau_{T1x} + \tau_{T2x} + \tau_{Mtx})^2 + 3(\tau_{T1y} + \tau_{T2y} + \tau_{Mty})^2}$$

# 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 and also for a detailed evaluation of stresses in a beam when the section forces to which it 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, and 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

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.

## MATERIAL PROPERTIES

Enter a value for **Poisson's ratio**  $\nu$  (dimensionless). The default is 0.3. This value only influences the detailed distribution of shear stresses caused by a transversal load.

## SECTION FORCES

If you want to perform a detailed computation of the stress state in a beam cross section, enter the section forces.



After changing data in this section, you do not need to compute the study for this physics interface again once it has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.

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**  $M_t$ .

## TWIST

Enter a value or expression for the **Axial twist**,  $\phi_{x1}$ . The twist is used to compute the axial warping displacement.

## DISCRETIZATION

Select an **Element order** — **Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**.

## Hole

Use the **Hole** feature to define internal holes for the beam cross section. One **Hole** feature must be added for each internal hole.

## 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>Hole**

*Ribbon*

Physics tab with **Beam Cross Section** interface selected:

**Boundaries>Beam Cross Section>Hole**

*Safety*

Use the **Safety** node 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 will be defined, as described in [Table 9-3](#). You can add any number of **Safety** nodes.

TABLE 9-3: 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

## FAILURE MODEL

Select a **Failure Criterion** — **von Mises Isotropic**, **Tresca Isotropic**, or **Rankine Isotropic**.

- When **Failure Criterion** is **von Mises Isotropic**, enter **Tensile strength**  $\sigma_{ts}$ .
- When **Failure Criterion** is **Tresca Isotropic**, enter **Tensile strength**  $\sigma_{ts}$ .
- When **Failure Criterion** is **Rankine Isotropic**, enter **Tensile strength**  $\sigma_{ts}$  and **Compressive strength**  $\sigma_{cs}$ .
- When **Failure Criterion** is **Bresler-Pister Isotropic**, enter **Tensile strength**  $\sigma_{ts}$ , **Compressive strength**  $\sigma_{cs}$ , and **Biaxial compressive strength**  $\sigma_{bc}$ .
- When **Failure Criterion** is **Willam-Warnke Isotropic**, enter **Tensile strength**  $\sigma_{ts}$ , **Compressive strength**  $\sigma_{cs}$ , and **Biaxial compressive strength**  $\sigma_{bc}$ .

- When **Failure Criterion** is **Ottosen Isotropic**, enter the **Compressive strength**  $\sigma_{CS}$ , **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>Safety**


*Ribbon*

Physics tab with **Beam Cross Section** selected:

**Domains>Safety**

# 10

## Truss

This chapter describes the Truss interface, which is found under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Modeling with Truss Elements](#)
- [Theory for the Truss Interface](#)
- [The Truss Interface](#)

# 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
- 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

The Truss 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 can use the Truss 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:

- It is generally recommended that you use first order shape functions when modeling wire-like structures. This is the default in the Truss interface.
- 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  $E_0 * (1 - 0.9999 * (\text{truss.en} < 0))$ .

- 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.

# Theory for the Truss Interface

## *Theory Background for the Truss Interface*

---

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  $\epsilon_n$  is calculated by expressing the global strains in tangential derivatives and projecting the global strains on the edge.

$$\epsilon_n = \mathbf{t}^t \epsilon_{gT} \mathbf{t}$$

where  $\mathbf{t}$  is the edge tangent vector and  $\epsilon_{gT}$  is defined as

$$\epsilon_{gT} = \begin{bmatrix} \epsilon_{xT} & \epsilon_{xyT} & \epsilon_{xzT} \\ \epsilon_{xyT} & \epsilon_{yT} & \epsilon_{yzT} \\ \epsilon_{xzT} & \epsilon_{yzT} & \epsilon_{zT} \end{bmatrix}$$

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

$$\epsilon_{ijT} = \frac{1}{2} \left( \left. \frac{\partial u_i}{\partial x_j} \right|_T + \left. \frac{\partial u_j}{\partial x_i} \right|_T + \left. \frac{\partial u_k}{\partial x_i} \right|_T \cdot \left. \frac{\partial u_k}{\partial x_j} \right|_T \right)$$

The engineering strain tensor used for small displacements is defined as

$$\epsilon_{ijT} = \frac{1}{2} \left( \left. \frac{\partial u_i}{\partial x_j} \right|_T + \left. \frac{\partial u_j}{\partial x_i} \right|_T \right) \quad (10-1)$$

The axial strain written out becomes

$$\begin{aligned}\varepsilon_n = & t_x(\varepsilon_x T t_x + \varepsilon_{xy} T t_y + \varepsilon_{xz} T t_z) + \\ & t_y(\varepsilon_{xy} T t_x + \varepsilon_y T t_y + \varepsilon_{yz} T t_z) + \\ & t_z(\varepsilon_{xz} T t_x + \varepsilon_{yz} T t_y + \varepsilon_z T t_z)\end{aligned}$$



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,  $\sigma_n$ , 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
- $\varepsilon_n$  is the total axial strain
- $\varepsilon_{n,el}$  is the elastic axial strain
- $\varepsilon_{inel}$  is the sum of all inelastic strain contributions:
  - Initial strain,  $\varepsilon_0$
  - Thermal strain,  $\varepsilon_{th}$
  - Hygroscopic strain,  $\varepsilon_{hs}$
  - Plastic strain,  $\varepsilon_{pl}$
- $\sigma_{ex}$  is the sum of all extra strain contributions:
  - Initial stress  $\sigma_0$
  - External stress  $\sigma_{ext}$

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  $\nu$ . 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  $\nu$ .

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

$$\delta W = \int_L (-\varepsilon_{ntest} \sigma_n A + \mathbf{u}_{test}^t \mathbf{F}_L) dL + \sum_i \mathbf{u}_{test}^t \mathbf{F}_{Pi}$$



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

$$\mathbf{x}_{di} = \mathbf{u}_i + \mathbf{x}_i \quad (10-2)$$

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

$$\mathbf{x} + \mathbf{u} = \mathbf{x}_{d1} + t\mathbf{a} \quad (10-3)$$

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 10-3 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

$$\mathbf{u} = \frac{\mathbf{u}_1(x_{n2} - x_n) + \mathbf{u}_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} + u_{ax}(\mathbf{x}_2 - \mathbf{x}_1) \quad (10-4)$$

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 10-4 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:

$$\begin{aligned} & \left[ \frac{u_1(x_{n2} - x_n) + u_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - u \right] (z_2 - z_1) - \\ & \left[ \frac{w_1(x_{n2} - x_n) + w_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - w \right] (x_2 - x_1) = 0 \\ & \left[ \frac{v_1(x_{n2} - x_n) + v_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - v \right] (z_2 - z_1) - \\ & \left[ \frac{w_1(x_{n2} - x_n) + w_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - w \right] (y_2 - y_1) = 0 \\ & \left[ \frac{v_1(x_{n2} - x_n) + v_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - v \right] (x_2 - x_1) - \\ & \left[ \frac{u_1(x_{n2} - x_n) + u_2(x_n - x_{n1})}{(x_{n2} - x_{n1})} - u \right] (y_2 - y_1) = 0 \end{aligned} \quad (10-5)$$

## 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.

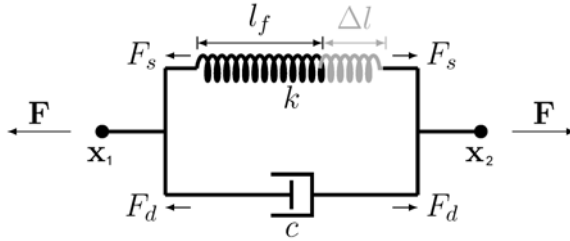


Figure 10-1: Conceptual sketch of the Spring-Damper Material.

#### 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  $\mathbf{X}_2$  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  $\Delta l_0$ , so that the free length of the spring is

$$l_f = l_0 - \Delta l_0$$

You can also specify the free length of the spring explicitly.

The spring extension  $\Delta l$  is computed as the difference between the current spring length and the free length,

$$\Delta l = l - l_f$$

#### 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  $k$ :

$$F_s = k\Delta l$$

If  $k$  depends on the extension, so that the spring is nonlinear, it should be interpreted as a secant stiffness, that is

$$F_s = k(\Delta l) \cdot \Delta l$$

You can also 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 `truss.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. In frequency domain, it is also possible to specify a loss factor  $\eta$ , and the total damping force will then be

$$F_d = i\omega c\Delta l + i\eta F_s$$

The magnitude of the total force is

$$F = F_s + F_d$$

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

$$W_s = \frac{1}{4} i_{ac} \text{Re}(\text{lindev}(F_s) \cdot \text{conj}(\text{lindev}(\Delta l)))$$

$$W_d = \frac{1}{4} i_{ac} \text{Re}(\text{lindev}(F_d) \cdot \text{conj}(\text{lindev}(\Delta l)))$$

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

$$F_c = \frac{\pi^2 EI_{\min}}{(KL)^2} \quad (10-6)$$

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 non-ideal 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:



$$f_i = \frac{-N}{F_c} \quad (10-7)$$

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 10-1](#).

TABLE 10-1: VARIABLES FOR LOCAL BUCKLING

	DEFINITION	DESCRIPTION
truss.lbFc	<a href="#">Equation 10-6</a>	Local buckling critical force
truss.lbf_i	<a href="#">Equation 10-7</a>	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

# 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 <https://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`.

## REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this

reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

### 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 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  $\mathbf{u}$**  which has two components ( $u, v$ ) in 2D and three components ( $u, v, w$ ) 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](#)
- [Edge Load](#)
- [Theory for the Truss Interface](#)



- *Vibrating String*: Application Library path  
**Structural\_Mechanics\_Module/Verification\_Examples/vibrating\_string**
  - *In-Plane and Space Truss*: Application Library path  
**Structural\_Mechanics\_Module/Verification\_Examples/  
inplane\_and\_space\_truss**
-



## *Boundary, Edge, Point, and Pair Nodes for the Truss Interface*

---

The [Truss 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 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
- Shape Memory Alloy
- Pinned
- Point Mass
- Point Mass Damping
- Phase
- 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 or Beam interface:

- Activation
- Added Mass
- Average Rotation
- Damping
- Free
- Gravity<sup>1</sup>
- Initial Values
- Plasticity
- Point Load
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rotating Frame<sup>1</sup>
- Spring Foundation
- Spring-Damper

<sup>1</sup> 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 DATA**

Enter the cross-section **Area**  $A$ .



This is required input data.

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.

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**

*Ribbon*

Physics tab with **Truss** 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 will make the model singular because the truss element only has stiffness in the axial direction in a geometrically linear problem. The same problem is present also 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.

### **CONSTRAINT ACTIVATION**

Select the **Disable for linear discretization order** check box to suppress the addition of the straight edge constraints when linear shape functions are used. This is the default state.



See also [Theory for Straight Edge Constraint](#) and [Modeling Wires and Cables](#).

---

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Truss>Line Constraints>Straight Edge Constraint**

#### *Ribbon*

Physics tab with **Truss** 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**

In this section there is always one check box. Either **Force linear strains** or **Include geometric nonlinearity** is shown.

If a study step is geometrically nonlinear, the default behavior is to use a large strain formulation for all boundaries or edges. There are however some cases when you would still want to use a small strain formulation in a part of the structure. In those cases, select the **Force linear strains** check box. When selected, a small strain formulation is always used, independently of the setting in the study step. The default value is that the check box is cleared, except when opening a model created in a version prior to 4.3. In this case the state is chosen so that the properties of the model are conserved.

The **Include geometric nonlinearity** check box is displayed only if the model was created in a version prior to 4.3, and geometric nonlinearity was originally used for the selected boundaries or edges. It is then selected and forces the **Include geometric nonlinearity** check box in the study step to be selected. If the check box is cleared, the check box is permanently removed and the study step assumes control over the selection of geometric nonlinearity.

#### **ENERGY DISSIPATION**

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

Select the **Calculate dissipated energy** check box as needed to compute the energy dissipated by **Plasticity**.

#### LOCATION IN USER INTERFACE

##### *Context Menus*

**Truss>Material Models>Linear Elastic Material**

##### *Ribbon*

Physics tab with **Truss** 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 Model** 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**

##### *Ribbon*

Physics tab with **Plasticity** node selected in the model tree:

**Attributes>Set Variables**

## *Shape Memory Alloy*

---

The **Shape Memory Alloy** feature is used to model stress-strain relationships which are nonlinear even at infinitesimal strains. This material model requires the Nonlinear Structural Materials Module.

By adding the following subnodes to the **Shape Memory Alloy** node you can incorporate other effects:

- [Thermal Expansion \(for Shape Memory Alloy\)](#)
- [Initial Stress and Strain](#)
- [External Stress](#)

Note: Some options are only available with certain COMSOL products (see <https://www.comsol.com/products/specifications/>)



The settings for this feature are documented in the section [Shape Memory Alloy](#) in the *Solid Mechanics* chapter. See also the section [Shape Memory Alloy](#) in the *Structural Mechanics Theory* chapter.

---

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Solid Mechanics>Material Models>Shape Memory Alloy**  
**Truss>Material Models>Shape Memory Alloy**

#### *Ribbon*

Physics tab with **Truss** selected:

**Edges>Material Models>Shape Memory Alloy**

## *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**  $T_{\text{ref}}$  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

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

where  $\alpha$  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

$$\epsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where  $\alpha_t$  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**

*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**  $T_{\text{ref}}$  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  $\xi_{\text{A}}$  and  $\xi_{\text{M}}$  are the austenite and martensite volume fractions. The default **Secant coefficient of thermal expansion for Austenite**  $\alpha_{\text{A}}$  and **Secant coefficient of thermal expansion for Martensite**  $\alpha_{\text{M}}$  use values **From material**.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_{\text{t}}(\tau) d\tau\right) - 1$$

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**  $\alpha_{tA}$  and **Tangent coefficient of thermal expansion for Martensite**  $\alpha_{tM}$  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

$$\epsilon_{hs} = \beta_h (c_{mo} - c_{mo,ref})$$

where  $\beta_h$  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**

*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**  $\sigma_{ni}$ .



For details about initial stresses and strains, see [Inelastic Strain Contributions](#) and [Initial Stresses and Strains](#).

### **LOCATION IN USER INTERFACE**

*Context Menus*

**Truss>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**  $S_{\text{ext}}$  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** or **Load contribution only** to determine the effect of the contribution.
- When **Axial stress** is selected, you enter a value or an expression for the **Axial stress**  $S_{n,\text{ext}}$ . Choose a **Contribution type** — **Add to stress tensor** or **Load contribution only** 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**

## 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>.d1`, for example `truss.spd1.d1`. The default expression is  $(1[\text{N/m}]) * \text{truss.spd1.d1}$ , 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**  $\eta$ .

### **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  $\Delta l_0$ . 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** check box to enter a **Deactivation expression**  $i_{\text{dac}}$ . 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** check box 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.d1>0.12[m]`, and select the **Permanently deactivate** check box. If the check box 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 box.



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)

*Ribbon*

Physics tab with **Truss** 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  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  as

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

## 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**  $\alpha$ . 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  $\alpha$ .
- A value or expression for the **Volume reference temperature**  $T_{\text{ref}}$  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**

### *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 box.

---

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<p><a href="#">Symmetry Constraints</a></p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## LOCATION IN USER INTERFACE

### *Context Menus*

**Truss>Line Constraints>Symmetry** (Line)

**Truss>More Constraints>Symmetry** (Point)

### *Ribbon*

Physics tab with **Truss** 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.

## CONSTRAINT SETTINGS

To display this section, click the **Show More Options** button (  ) and select **Advanced Physics Options** in the **Show More Options** dialog box.

---

	In the <i>COMSOL Multiphysics Reference Manual</i> : <ul style="list-style-type: none"><li>• <a href="#">Constraint Reaction Terms</a></li><li>• <a href="#">Weak Constraints</a></li><li>• <a href="#">Constraint Settings</a></li></ul>
	<a href="#">Symmetry Constraints</a>
	You can activate and deactivate this boundary condition by assigning it to a constraint group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.

---

## LOCATION IN USER INTERFACE

### Context Menus

**Truss>Line Constraints>Antisymmetry** (Line)

**Truss>More Constraints>Antisymmetry** (Point)

### Ribbon

Physics tab with **Truss** 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).

### FORCE

Select a **Load type** — **Force per unit length**, **Force per unit volume**, or **Total force**. Enter values or expressions for the components.

TABLE 10-2:

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit length	$F_L$	N/m	edges	3D (x, y, z)
			boundaries	2D (x, y)
Force per unit volume	$F_V$	N/m <sup>3</sup>	edges	3D (x, y, z)
			boundaries	2D (x, y)
Total force	$F_{tot}$	N	edges	3D (x, y, z)
			boundaries	2D (x, y)



- When **Force per unit 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.
- 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**

*Ribbon*

Physics tab with **Truss** 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.

The [Point Mass Damping](#) subnode can be added to specify a mass-proportional damping.

#### **POINT MASS**

Enter a **Point mass**  $m$ .

#### **LOCATION IN USER INTERFACE**

*Context Menus*

**Truss>Mass, Spring, and Damper>Point Mass**

*Ribbon*

Physics tab with **Truss** selected:

**Edges>Mass, Spring, and Damper>Point Mass**

*Point Mass Damping*

---

Use the **Point Mass Damping** subnode to add damping to a [Point Mass](#) parent node.

#### **POINT MASS DAMPING**

Enter a **Mass damping parameter**  $\alpha_{dM}$ . This is the mass proportional term of a Rayleigh damping.

#### **LOCATION IN USER INTERFACE**

*Context Menus*

**Truss>Point Mass>Point Mass Damping**

*Ribbon*

Physics tab with **Point Mass** node selected in the model tree:

**Point Mass Damping**

## *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 ( $\phi$ ), 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 load  $\mathbf{F}_{\text{ph}}$  for harmonic loads. Enter the phase for each component of the load in the corresponding fields.

### **LOCATION IN USER INTERFACE**

#### *Context Menus*

**Truss>Edge Load>Phase**

**Truss>Point Load>Phase**


#### *Ribbon*

Physics tab with **Edge Load** or **Point Load** node selected in the model tree:

**Attributes>Phase**



# Pipe Mechanics

This chapter describes the Pipe Mechanics interface, which you find under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Theory for the Pipe Mechanics Interface](#)
- [The Pipe Mechanics Interface](#)

# 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 documentation for the Beam interface:

- [Shape Functions](#)
- [Geometric Nonlinearity](#)
- [Strain-Displacement/Rotation Relation](#)
- [Stress-Strain Relation](#)
- [Initial Load 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 *overpressure*, 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 overpressure

$$p = P_{\text{internal}} - P_{\text{external}}$$

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  $\gamma_m$  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 11-1).

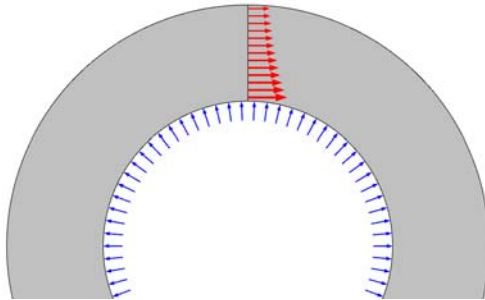


Figure 11-1: Hoop stress distribution in a thick-walled pipe.

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

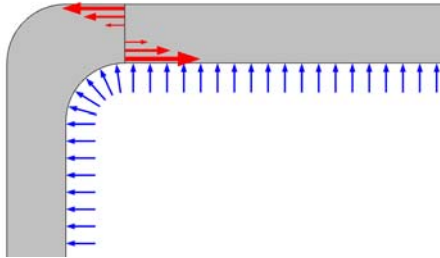
$$\sigma_{h, b} = \frac{p}{1 + \frac{d_i}{d_o}} = \gamma_b p$$

Here  $\gamma_b$  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 11-2](#).



*Figure 11-2: 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

$$\sigma_{h,b} = \frac{6M}{t^2} = \frac{p}{2t^2} \cdot \frac{L_y^3 + L_z^3}{L_y + L_z}$$



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

$$\sigma_{h,m} = p \frac{\max(L_y, L_z)}{t}$$



The average hoop stress is available in the variable `pipem.shm`.

Thus,

$$\gamma_m = \frac{\max(L_y, L_z)}{t}$$

$$\gamma_b = \frac{1}{2t^2} \cdot \frac{L_y^3 + L_z^3}{L_y + L_z}$$

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 actually be shown 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

$$\epsilon_{ap} = \frac{-2\nu p A_f}{EA}$$

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 on the Fluid Properties feature under Pipe Flow interface. For most friction models, it is effectively proportional to  $1/\text{Re}$ , where  $\text{Re} = \rho_f |u_f| d_h / \mu_f$  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

$$\mathbf{F}_{jp} = \sum_n s_n [(p_f - p_o + \rho_f u_f^2) A_f \mathbf{t}]_n$$

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.



One needs 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 has 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

$$T_m = \frac{T_i + T_o}{2}$$

If the coefficient of thermal expansion is temperature dependent, it is assumed that its variation through the pipe wall is linear.

Define

$$\begin{aligned}\alpha_i &= \alpha(T_i) \\ \alpha_o &= \alpha(T_o) \\ \Delta\alpha &= \alpha_o - \alpha_i \\ \Delta T &= T_o - T_i\end{aligned}$$

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



$$\begin{aligned} \varepsilon_{th} = & \alpha_i(T_i - T_{ref}) + \frac{1}{3}\Delta\alpha(T_i - T_{ref})\left(1 + \frac{r_o}{r_o + r_i}\right) + \\ & \alpha_i\Delta T\left(\frac{r_o^2}{r_o^2 - r_i^2} - \frac{1}{2\log\left(\frac{r_o}{r_i}\right)}\right) + \\ & \Delta\alpha\Delta T\left(\frac{r_o^2(2r_o - 3r_i)}{3(r_o - r_i)^2(r_o + r_i)} - \frac{4r_o^2 - 5r_i^2 - 5r_or_i}{18(r_o^2 - r_i^2)\log\left(\frac{r_o}{r_i}\right)}\right) \end{aligned}$$

### 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

$$\sigma_{t,nb} = \sigma_{t,hb} = \frac{(\alpha_i + \alpha_o)\Delta T}{2(1 - \nu)} \left[ \frac{r_o^2}{r_o^2 - r_i^2} - \frac{1}{2\log\left(\frac{r_o}{r_i}\right)} \right]$$

### *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  $(y_{1k}, z_{1k})$  in the cross section as

$$\sigma_{bk} = \frac{M_{1y}z_{1k}}{I_{yy}} - \frac{M_{1z}y_{1k}}{I_{zz}}$$

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 overpressure 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

$$\sigma_k = \sigma_{bk} + \sigma_n + (\sigma_{n,b} + \sigma_{t,b})\text{sign}(\sigma_{bk} + \sigma_n)$$

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

$$\sigma_{b, \max} = \frac{d_o \sqrt{M_{1y}^2 + M_{1z}^2}}{2I_{zz}}$$

where  $d_o$  is the outer diameter. This value replaces the stress from the stress evaluation points ( $\sigma_{bk}$ ) 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,  $\sigma_n$ : `pipem.sn`
- Stress caused by bending moments,  $\sigma_{bk}$  (evaluation points 1-4): `pipem.sb1`, ..., `pipem.sb4`
- Axial stress from overpressure,  $\sigma_{n,b}$ : `pipem.shb`
- Axial stress from temperature gradient,  $\sigma_{t,b}$ : `pipem.shb`
- Total axial stress in evaluation points,  $\sigma_k$ : `pipem.s1`, ..., `pipem.sb4`
- Maximum axial stress,  $\sigma_{max}$ : `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  $\gamma_m$  and  $\gamma_b$  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 overpressure, 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

$$\tau_{t, \max} = \frac{|M_{1x}|}{W_t}$$

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.

$$T_{1z} = EI_{yy} \frac{\partial^2 \theta_{1y}}{\partial s^2}$$
$$T_{1y} = -EI_{zz} \frac{\partial^2 \theta_{1z}}{\partial s^2}$$

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

$$\tau_{sz, \text{ave}} = \frac{T_{1z}}{A}$$
$$\tau_{sy, \text{ave}} = \frac{T_{1y}}{A}$$
(11-1)

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\tau_{sz, \max} = \mu_z \tau_{sz, \text{ave}}$$
$$\tau_{sy, \max} = \mu_y \tau_{sy, \text{ave}}$$
(11-2)

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

$$\tau_{xz, \max} = |\tau_{sz, \max}| + \tau_{t, \max}$$
$$\tau_{xy, \max} = |\tau_{sy, \max}| + \tau_{t, \max}$$

## EQUIVALENT STRESS

The maximum von Mises equivalent stress for the cross section is then defined as

$$\sigma_{\text{mises}} = \sqrt{\sigma_{n, \text{max}}^2 + \sigma_{h, \text{max}}^2 + 3\tau_{xy, \text{max}}^2 + 3\tau_{xz, \text{max}}^2}$$

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 non-zero principal stresses in that plane are computed for evaluation point  $k$  as

$$\sigma_{1,2}^k = \frac{1}{2}[\sigma_k + \sigma_{h, \text{max}} \pm \sqrt{(\sigma_k - \sigma_{h, \text{max}})^2 + 4(\tau_{xy, \text{max}}^2 + \tau_{xz, \text{max}}^2)}]$$

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,  $\sigma_{\text{mises}}$ : `pipem.mises`
- Tresca equivalent stress,  $\sigma_{\text{tresca}}$ : `pipem.tresca`
- First principal stress,  $\sigma_{1k}$  (evaluation points 1-4): `pipem.sp1_1`, ..., `pipem.sp1_4`
- First principal stress,  $\sigma_{2k}$  (evaluation points 1-4): `pipem.sp2_1`, ..., `pipem.sp2_4`
- First principal stress,  $\sigma_{3k}$  (evaluation points 1-4): `pipem.sp3_1`, ..., `pipem.sp3_4`

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 Solids](#) and [Connection Between Shells and Beams](#) in the Shell documentation.

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

$$\mathbf{u}_s = \mathbf{u}_p + \theta_p \times \mathbf{r} - \nu_s \left( \int \epsilon_{x1,s} dy_1 \right) \mathbf{e}_{y1} - \nu_s \left( \int \epsilon_{x1,s} dz_1 \right) \mathbf{e}_{z1} - \mathbf{u}_{\text{radial}} \quad (11-3)$$

where  $\mathbf{u}_s$  is the displacement of the solid,  $\nu_s$  is Poisson's ratio of the solid,  $\mathbf{u}_p$  is the displacement of the pipe,  $\theta_p$  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 11-3](#) 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  $\mathbf{u}_{\text{radial}}$  included in [Equation 11-3](#) is computed from the radial deformation  $u_r$  of the pipe as

$$\mathbf{u}_{\text{radial}} = \mathbf{T}^T \cdot \left\{ 0, \frac{u_r d_y}{R}, \frac{u_r d_z}{R} \right\}$$

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

$$u_r = \frac{R}{E_s} [S_\theta - \nu_s (S_R + S_Z)] + R \alpha_p \left[ T_{\text{in}} - T_{\text{ref}} + (T_{\text{out}} - T_{\text{in}}) \frac{\log\left(\frac{R}{R_i}\right)}{\log\left(\frac{R_o}{R_i}\right)} \right] \quad (11-4)$$

where  $E_s$  is Young's modulus of the solid,  $\alpha_p$  is its coefficient of thermal expansion,  $R_i$  and  $R_o$  are its inner and outer radii, and  $T_{\text{ref}}$  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

$$\begin{aligned} \theta_p \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 &= 0 \\ \theta_p \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 &= 0 \end{aligned}$$

where  $\mathbf{a}$  is the shell normal displacement, and  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are the shell tangents. Also, the expression for the radial deformation  $u_r$  in Equation 11-4 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

$$S_R = 0$$

$$S_{\theta} = s_p \frac{pR_i}{d} + s_T(T_{\text{out}} - T_{\text{in}})\alpha \frac{E_s}{2(1-\nu_s)} \left[ \frac{2}{1 - \left(\frac{R_i}{R_o}\right)^2} - \frac{1}{\log\left(\frac{R_o}{R_i}\right)} \right]$$

$$S_Z = s_T(T_{\text{out}} - T_{\text{in}})\alpha \frac{E_s}{2(1-\nu_s)} \left[ \frac{2}{1 - \left(\frac{R_i}{R_o}\right)^2} - \frac{1}{\log\left(\frac{R_o}{R_i}\right)} \right]$$

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

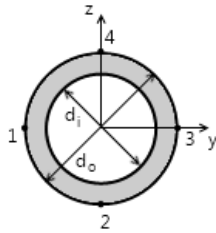


Figure 11-3: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 11-1: 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 11-1: CIRCULAR SECTION CONSTANTS

PROPERTY	FORMULA	REMARKS
$e_z$	0	
$\mu_y$	$\frac{(d_o^3 - d_i^3)A}{12(d_o - d_i)I_{zz}}$	
$\kappa_y$	$1 + \frac{d_i}{d_o}$	
$I_{yy}$	$I_{zz}$	
$e_y$	0	
$\mu_z$	$\mu_y$	
$\kappa_z$	$\kappa_y$	
$J$	$\frac{\pi(d_o^4 - d_i^4)}{32}$	
$W_t$	$\frac{\pi(d_o^4 - d_i^4)}{16d_o}$	
$p_1$	$(\frac{-d_o}{2}, 0)$	
$p_2$	$(0, \frac{-d_o}{2})$	
$p_3$	$(\frac{d_o}{2}, 0)$	
$p_4$	$(0, \frac{d_o}{2})$	
$r_e$	$\frac{d_o}{2}$	

## SQUARE SECTION

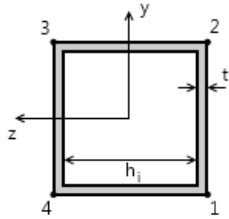


Figure 11-4: Geometry of a square shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 11-2: 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	
$\mu_y$	$\frac{((h_i + 2t)^3 - h_i^3) A}{16 t I_{zz}}$	
$\kappa_y$	$\frac{2 (h_i + 2t) t_z}{A}$	
$I_{yy}$	$I_{zz}$	
$e_y$	0	
$\mu_z$	$\mu_y$	
$\kappa_z$	$\kappa_y$	
$J$	$\frac{2 t (h_i + t)^4}{h_i + t}$	Thin-walled approximation
$W_t$	$2 t (h_i + t)^2$	Thin-walled approximation

TABLE 11-2: 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**

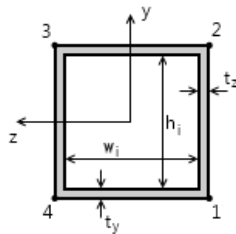


Figure 11-5: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.



TABLE 11-3: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$A$	$2(h_i t_z + w_i t_y) + 4t_y t_z$	
$I_{zz}$	$\frac{t_z h_i^3 + t_y^3 (w_i + 2t_z)}{6} + \frac{t_y (w_i + 2t_z) (h_i + t_y)^2}{2}$	
$e_z$	0	

TABLE 11-3: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
$\mu_y$	$\frac{((h_i + 2t_y)^2(w_i + 2t_z) - h_i^2 w_i)A}{16t_z I_{zz}}$	
$\kappa_y$	$\frac{2(h_i + 2t_y)t_z}{A}$	
$I_{yy}$	$\frac{t_y w_i^3 + t_z^3 (h_i + 2t_y)}{6} + \frac{t_z (h_i + 2t_y)(w_i + t_z)^2}{2}$	
$e_y$	0	
$\mu_z$	$\frac{((w_i + 2t_z)^2(h_i + 2t_y) - w_i^2 h_i)A}{16t_y I_{yy}}$	
$\kappa_z$	$\frac{2(w_i + 2t_z)t_y}{A}$	
$J$	$\frac{2(h_i + t_y)^2 (w_i + t_z)^2}{\frac{h_i + t_y}{t_z} + \frac{w_i + t_z}{t_y}}$	Thin-walled approximation
$W_t$	$2(h_i + t_y)(w_i + t_z) \cdot \min(t_y, t_z)$	Thin-walled approximation
$p_1$	$\left(-\frac{h_i + 2t_y}{2}, -\frac{w_i + 2t_z}{2}\right)$	
$p_2$	$\left(\frac{h_i + 2t_y}{2}, -\frac{w_i + 2t_z}{2}\right)$	
$p_3$	$\left(\frac{h_i + 2t_y}{2}, \frac{w_i + 2t_z}{2}\right)$	
$p_4$	$\left(-\frac{h_i + 2t_y}{2}, \frac{w_i + 2t_z}{2}\right)$	
$r_e$	$\frac{h_i + 2t_y + w_i + 2t_z}{4}$	

# 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.

#### **REFERENCE POINT FOR MOMENT COMPUTATION**

Enter the default coordinates for the **Reference point for moment computation**  $\mathbf{x}_{\text{ref}}$ . The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

#### **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  $\mathbf{u}$ , which has two components ( $u$ ,  $v$ ) in 2D and three components ( $u$ ,  $v$ , and  $w$ ) in 3D.
- The rotation angle  $\theta$ , 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 through 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):

- [Fluid and Pipe Materials](#)
- [Fluid Load](#)
- [Gravity](#)
- [Pipe Cross Section](#)
- [Rigid Domain](#)
- [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 Mass
- Point Mass Damping
- 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**  $d_i$

#### *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	$I_{zz}$	$m^4$
2D and 3D, Timoshenko beam	Shear correction factor along local y-axis	$\kappa_y$	l
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	$\kappa_z$	l



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 11-1](#) and [Equation 11-2](#).

#### *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	$\mu_y$	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	$\mu_z$	l
2D and 3D	Pressure membrane stress factor	$\gamma_m$	l
2D and 3D	Pressure bending stress factor	$\gamma_b$	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	$\mu_y$	l
3D only	Torsional section modulus	$W_t$	$m^3$
3D only	Max shear stress factor in local z direction	$\mu_z$	l
2D and 3D	Pressure membrane stress factor	$\gamma_m$	l
2D and 3D	Pressure bending stress factor	$\gamma_b$	l

## 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 ( $\mathbf{e}_{x1}$ ,  $\mathbf{e}_{y1}$ ,  $\mathbf{e}_{z1}$ ) is formed using the following algorithm:

$$\mathbf{v}_{z1} = \mathbf{e}_{x1} \times (\mathbf{p} - \mathbf{m})$$

$$\mathbf{e}_{z1} = \frac{\mathbf{v}_{z1}}{|\mathbf{v}_{z1}|}$$

$$\mathbf{e}_{y1} = \mathbf{e}_{z1} \times \mathbf{e}_{x1}$$

Here,  $\mathbf{p}$  is the reference point, and  $\mathbf{m}$  is the midpoint of the pipe element. The definition of the local coordinate system is illustrated in Figure 11-6.

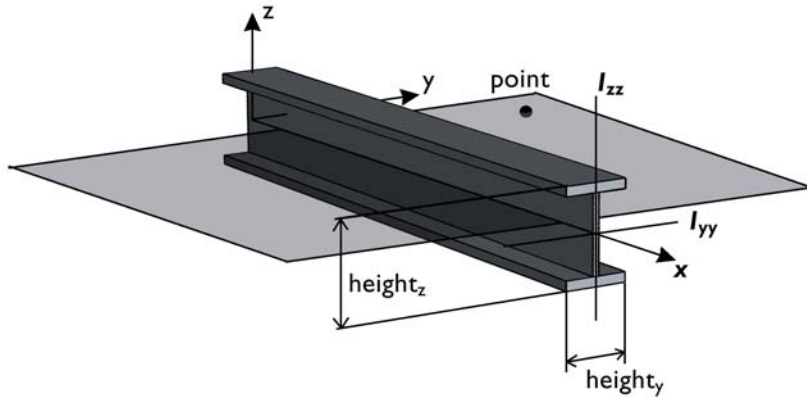


Figure 11-6: Local beam coordinate system defined by a reference point.

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  $\phi$** . 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 ( $\mathbf{e}_{x1}$ ,  $\mathbf{e}_{y1}$ ,  $\mathbf{e}_{z1}$ ) is formed using the following algorithm:

$$\mathbf{v}_{z1} = \mathbf{e}_{x1} \times \mathbf{V}$$

$$\mathbf{e}_{z1} = \frac{\mathbf{v}_{z1}}{|\mathbf{v}_{z1}|}$$

$$\mathbf{e}_{y1} = \mathbf{e}_{z1} \times \mathbf{e}_{x1}$$

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**

### *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**,  $\rho_f$ , 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  $\nu$ .
- **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  $\lambda$  and  $\mu$ .
- **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**  $\rho$  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](#)).

### **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 Domain*

Add the **Rigid Domain** node and select one or more edges (2D: boundaries) to make them a rigid body. **Rigid Domain** 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 \(Rigid Domain\)](#)).

You can add functionality to the rigid domain through the following subnodes:

- [Fixed Constraint \(Rigid Domain\)](#) to fully constrain the rigid domain.
- [Prescribed Displacement/Rotation](#) to prescribe the displacement of individual degrees of freedom.
- [Applied Force \(Rigid Domain\)](#) to apply a force in given point.
- [Applied Moment \(Rigid Domain\)](#) to apply a moment.
- [Mass and Moment of Inertia \(Rigid Domain\)](#) to add extra mass and moment of inertia in a given point.
- [Spring Foundation \(Rigid Domain\)](#) 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**,  $\rho_f$ , 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**,  $\rho$ , 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 Domain** node are described in the documentation for [Rigid Domain](#) in the Solid Mechanics interface.

## LOCATION IN USER INTERFACE

*Context Menus*

**Pipe Mechanics>Material Models>Rigid Domain**

*Ribbon*

Physics tab with **Pipe Mechanics** selected:

**Edges>Material Models>Rigid Domain (3D)**

**Boundaries>Material Models>Rigid Domain (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)  $\alpha$ , the temperature  $T$ , and the strain-free reference temperature  $T_{\text{ref}}$  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 internal temperature 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**  $T_{\text{ref}}$  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**  $\alpha$  list, select **From material** to use the coefficient of thermal expansion from a material, or **User defined** to enter a value or expression for  $\alpha$ . 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. You can then select the temperature degree of freedom from the drop-down menus, 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 **User defined** option, and enter the variables defined in the **Wall Layer** node 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 all selected features in the physics interface with a density, mass, or mass distribution. You select edges (in 2D, boundaries), and all points which 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 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). The action of gravity can also be seen as a linearly accelerated frame of reference.

## 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 gravitational forces.

## GRAVITY

Enter the components of the **Gravity g**. The default value is `g_const`, which is the physical constant having the value  $9.8066 \text{ m/s}^2$ .

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.



Rotating Frame and Gravity

---



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 **Gravity**.



- 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>Line and Volume Loads>Gravity**

### *Ribbon*

Physics tab with **Pipe Mechanics** selected:

**Edges>Line and Volume Loads>Gravity (3D)**

**Boundaries>Line and Volume Loads>Gravity (2D)**

### *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 overpressure is important, as discussed in [Effects of Internal Pressure](#). You can thus use two approaches:

- Enter the overpressure 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 unit 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 unit length** drop-down list can also contain drag force variables announced by the Pipe Flow interface.

Select a method for entering the **Centrifugal force per unit 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 unit length** drop-down list can also contain drag force variables announced by the Pipe Flow interface.

---

	<p>See also</p> <ul style="list-style-type: none"><li>• <a href="#">Effects of Internal Pressure</a> and <a href="#">Fluid Flow Forces</a> in the <a href="#">Theory for the Pipe Mechanics Interface</a> section</li><li>• <a href="#">Using Common Model Input</a></li><li>• <a href="#">Default Model Inputs</a> and <a href="#">Model Input</a> in the <i>COMSOL Multiphysics Reference Manual</i>.</li></ul>
	<p>You can assign this load to a load group. See <a href="#">Load Cases</a> in the <a href="#">Structural Mechanics Modeling</a> chapter.</p>

---

## 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)**



# 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](#)

- 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 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
- The Piezoelectricity Interface
- Piezoelectricity, Layered Shell Interface
- The Electrostriction Interface
- The Ferroelectroelasticity Interface
- The Magnetostriction Interface
- Hygroscopic Swelling Coupling
- Multiphysics Couplings
  - Thermal Expansion
  - Layered Thermal Expansion
  - Hygroscopic Swelling
  - Piezoelectric Effect
  - Layered Piezoelectric Effect
  - Electrostriction
  - Magnetostriction
  - Lorentz Coupling
  - Fluid-Structure Interaction
  - Fluid-Structure Interaction, Pair
  - Fluid-Structure Interaction, Fixed Geometry

- Couplings Between Structural Mechanics Interfaces
  - Solid-Thin Structure Connection
  - Solid-Beam Connection
  - Shell-Beam Connection
  - Layered Shell-Structure Transition
  - Layered Shell-Structure Cladding
  - Structure-Pipe Connection
  - Embedded Reinforcement

# 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.

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, which automatically includes the multiphysics coupling feature [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](#).



In previous versions of COMSOL Multiphysics, a specific physics interface called Thermal Stress 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*.



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 from the available coupling features, but the settings in the constituent interfaces are not modified.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

TABLE 12-1: MODIFIED SETTINGS FOR A THERMAL STRESS, SOLID INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under <a href="#">Structural Transient Behavior</a> the <b>Structural transient behavior</b> is set to <b>Quasi static</b> .
Thermal Expansion	The <b>Domain Selection</b> 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.

## PHYSICS INTERFACES AND COUPLING FEATURES


### *Coupling Features*


The [Thermal Expansion](#) is used to couple the interfaces.



- *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.

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 which automatically includes the multiphysics coupling feature **Layered Thermal Expansion**.

*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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---




TABLE 12-2: MODIFIED SETTINGS FOR A THERMAL STRESS, SHELL INTERFACE


PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Shell	For the Shell interface, under <a href="#">Structural Transient Behavior</a> the <b>Structural transient behavior</b> is set to <b>Quasi static</b> .
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to <b>Linear</b> .
Layered Thermal Expansion	The <b>Boundary Selection</b> 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.

### COUPLING FEATURES

The [Layered Thermal Expansion](#) is used to couple the interfaces.

# 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.

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 [Layered Thermal Expansion](#)

*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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.


TABLE 12-3: MODIFIED SETTINGS FOR A THERMAL STRESS, LAYERED SHELL INTERFACE


PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Layered Shell	For the Layered Shell interface, under <a href="#">Structural Transient Behavior</a> the <b>Structural transient behavior</b> is set to <b>Quasi static</b> .
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to <b>Linear</b> .
Layered Thermal Expansion	The <b>Boundary Selection</b> 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.

### COUPLING FEATURES

The [Layered Thermal Expansion](#) is used to couple the interfaces.

# 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.

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 [Layered Thermal Expansion](#).

*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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.


TABLE 12-4: MODIFIED SETTINGS FOR A THERMAL STRESS, MEMBRANE INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Membrane	For the Membrane interface, under <a href="#">Structural Transient Behavior</a> the <b>Structural transient behavior</b> is set to <b>Quasi static</b> .
Heat Transfer in Shells	For the Heat Transfer in Shell interface, under Discretization, the shape function order for the temperature is set to <b>Linear</b> .
Layered Thermal Expansion	The <b>Boundary Selection</b> 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.


### COUPLING FEATURES

The [Layered Thermal Expansion](#) is used to couple the interfaces.

# 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 field 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.

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, which automatically includes 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](#).



In previous versions of COMSOL Multiphysics, a specific physics interface called Joule Heating and Thermal Expansion 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 **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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

TABLE 12-5: 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 <b>Structural transient behavior</b> is set to <b>Quasi static</b> .
Electric Currents	No changes.
Heat Transfer in Solids	No changes.
Thermal Expansion	The <b>Domain Selection</b> 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 <b>Domain Selection</b> 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.

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help.

### *Coupling Features*

- The [Thermal Expansion](#) coupling feature node is described for [The Thermal Stress, Solid Interface](#).
- The [Electromagnetic Heating](#) multiphysics coupling nodes is described for [The Joule Heating Interface](#) in the *COMSOL Multiphysics Reference Manual*.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.


- The available physics features for [The Solid Mechanics Interface](#) are listed in the section [Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics](#).
- The available physics features for [The Heat Transfer in Solids Interface](#) are listed in the section [Feature Nodes for the Heat Transfer in Solids Interface](#).
- The available physics features for [The Electric Currents Interface](#) are listed in the section [Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface](#) in the *COMSOL Multiphysics Reference Manual*.



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 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 two-way coupling between solids and fluids. There are, however, special study steps available in order to model one-way fluid-structure interaction.

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, 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 Deforming Domain feature is described in the [Moving Mesh Features](#).
- 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 the FSI coupling.

---



---

In previous versions of COMSOL Multiphysics (up to version 5.3), a specific physics interface called **Fluid-Structure Interaction** 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, 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

### PHYSICS INTERFACES AND COUPLING FEATURES



---

Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

#### *Coupling Features*

The [Fluid-Structure Interaction](#) coupling feature is used to couple the interfaces.

### Physics Interface Features

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

---

- 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 Solid Mechanics Interface](#) are listed in the [Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics](#) 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, 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 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 Studies for Fluid-Structure 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### *Coupling Features*

The [Fluid-Structure Interaction](#) coupling feature is used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.




---

- 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 Solid Mechanics Interface](#) are listed in the [Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics](#) section.

# 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 two-way coupling between solids and fluids. There are, however, special study steps available in order to model one-way fluid-structure interaction.

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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES

---



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### *Coupling Features*

The [Fluid-Structure Interaction](#) coupling feature is used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

---

- 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 Studies for Fluid-Structure 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES

---



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### *Coupling Features*

The [Fluid-Structure Interaction](#) coupling feature is used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.




---

- 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-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 two-way coupling between solids and fluids. There are, however, special study steps available in order to model one-way fluid-structure interaction.

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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES

---



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### *Coupling Features*

The [Fluid-Structure Interaction](#) coupling feature is used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

---

- 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-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 Studies for Fluid-Structure 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.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES

---



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### *Coupling Features*

The [Fluid-Structure Interaction](#), [Fixed Geometry](#) coupling feature is used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

---

- 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 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 two-way coupling between solids and fluids. You can, however, by modifying the solver sequences also model various types of one-directional 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

#### PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

#### *Coupling Features*

The [Fluid-Structure Interaction](#), [Nonisothermal Flow](#), and [Thermal Expansion](#) coupling features are used to couple the interfaces.

### Physics Interface Features

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.




---

- 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 Solid Mechanics Interface](#) are listed in the [Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics](#) section.
- The available physics features for [The Heat Transfer in Solids and Fluids Interface](#) are listed in the [Domain Features](#), [Boundary Features](#), and [Edge and Point Features](#), sections.

# 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 two-way coupling between solids and the fluids. There are, however, special study steps available in order to model one-way fluid-structure interaction. 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

### PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

#### *Coupling Features*

[The Two-Phase Flow, Phase Field Coupling Feature](#) and the [Fluid-Structure Interaction](#) feature are used to couple the interfaces.


#### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

# 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 two-way coupling between solids and the fluids. There are, however, special study steps available in order to model one-way fluid-structure interaction. 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 Studies for Fluid-Structure 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 are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

### *Coupling Features*

[The Two-Phase Flow, Phase Field Coupling Feature](#) and the [Fluid-Structure Interaction](#) feature are used to couple the interfaces.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS 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.

---



# The Piezoelectricity Interface

The **Piezoelectricity** 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** 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 **Piezoelectric Effect**.

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 **Piezoelectric Effect** 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 Piezoelectric Effect 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 **Piezoelectric Effect** but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

## PHYSICS INTERFACES AND COUPLING FEATURES

---



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

### *Coupling Feature*

The [Piezoelectric Effect](#) 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. **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** feature is described for the Electrostatics interface.



- [Piezoelectric Material](#)
- [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 **Piezoelectric Effect** 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.

# Piezoelectricity, Layered Shell Interface

The **Piezoelectricity, Layered Shell** 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 **Layered Piezoelectric Effect**.

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 [Layered Piezoelectric Effect](#) 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 Layered Piezoelectric Effect 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 **Layered Piezoelectric Effect** but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

### PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

*Coupling Feature*

The [Layered Piezoelectric Effect](#) 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. **Coupling Loss**, **Dielectric Loss**, and **Mechanical Damping** subnodes are available for Piezoelectric Material under the Layered Shell interface. The **Piezoelectric Layer** feature is described for the Electric Currents in Layered Shells interface.



- [Piezoelectric Material](#)
- [Modeling Piezoelectric Problems](#)
-

# The Electrostriction Interface

The **Electrostriction** 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** interfaces are added to the **Model Builder**.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature [Electrostriction](#).

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 the **Charge Conservation** node with its selection set to all domains, and the material type set to Solid. This node serves to represent solid dielectric material domains. Any number of **Charge Conservation** nodes can be also added manually to the interface and configured to represent either solid or nonsolid (for example, air or free space) materials.



# The Ferroelectroelasticity Interface

The **Ferroelectroelasticity** 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** interfaces are added to the **Model Builder**.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature [Electrostriction](#).

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 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.

# The Magnetostriction Interface

The **Magnetostriction** 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 **Magnetostriction**.

The participating **Solid Mechanics** interface includes the default **Magnetostrictive Material** feature with its selection set to all domains. The **Magnetic Fields** interface gets a default **Ampère's Law, Magnetostrictive** feature with similar settings.

Such features can be also added manually to their corresponding interfaces similar to any other material models therein.

The **Magnetostriction** multiphysics coupling node can be active only on the selection, where both features Magnetostrictive Material and Ampère's Law, Magnetostrictive are active.

You input both the mechanical and magnetic material data under the Magnetostrictive Material, and the electrical properties under Ampère's Law, Magnetostrictive.

When it is used without an active Magnetostriction coupling feature, the Magnetostrictive Material node works similarly to a Linear Elastic Material feature with some limitations on the format for the elastic material data input. All the magnetic material data and coupling data will have no effect.

You use the Ampère's Law, Magnetostrictive feature under Magnetic Fields to select those 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, Magnetostrictive 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 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 nonsolid model for air (via Ampere’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 Magnetostriction 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 magnetostrictive 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 couplings, for example **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 **Magnetostriction** but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics Couplings** menu.

---

#### PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

---

### Coupling Feature

The [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. **Mechanical Damping** and **Thermal Expansion** subnodes are available for Magnetostrictive Material under the Solid Mechanics interface. The **Ampère's Law, Magnetostrictive** feature is described for the Magnetic Fields interface.



- [Magnetostrictive Material](#)
  - [Ampère's Law, Magnetostrictive](#)
  - [Modeling Magnetostrictive Materials](#)
- 



---

*Nonlinear Magnetostrictive Transducer*: Application Library path  
**Structural\_Mechanics\_Module/Magnetostrictive\_Devices/  
nonlinear\_magnetostriction**

---


# Hygroscopic Swelling Coupling

The **Hygroscopic Swelling** multiphysics coupling node is used 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. It is described in [Hygroscopic Swelling](#).

# 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 `te1`.

### 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 the 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**  $T_{\text{ref}}$  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  $\alpha$  is the secant coefficient of thermal expansion.  $\alpha$  can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where  $\alpha_t$  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 **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**.

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, and viscous damping.



---

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 box.

---






- Temperature-Dependent Material Data
- Thermoelastic Damping
- Entropy and Thermoelasticity

---

### *Layered Thermal Expansion*

---

Use the **Layered Thermal Expansion** 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 **Layered Thermal Expansion** node is only available with some COMSOL products (see <https://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 `Layered 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 `te11`.

#### **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 the 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 **Layered Thermal Expansion** 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 **Layered Thermal Expansion** 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 check boxes 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**  $T_{\text{ref}}$  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

$$\epsilon_{\text{th}} = \alpha(T - T_{\text{ref}})$$

where  $\alpha$  is the secant coefficient of thermal expansion.  $\alpha$  can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where  $\alpha_t$  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 **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**.

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, and viscous damping.




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 box.



- [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

$$\varepsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where  $\beta_h$  is the coefficient of hygroscopic swelling,  $c_{mo}$  is the moisture concentration, and  $c_{mo,ref}$  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 **Transport Properties**, **Porous Media Transport Properties**, or **Partially Saturated Porous Media** 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 the Hygroscopic Swelling 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**  $c_{mo,ref}$ . 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**  $\beta_h$  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  $\beta_h$ . The default value for the **User defined** case is  $1.5e-4 \text{ m}^3/\text{kg}$ . 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** check box 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.

### *Piezoelectric Effect*

The **Piezoelectric Effect** 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 **Piezoelectric Effect 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 the Piezoelectric Effect multiphysics coupling. The **Solid mechanics** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the Piezoelectric Effect 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 Piezoelectric Effect 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*.

### *Layered Piezoelectric Effect*

The **Layered Piezoelectric Effect** 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 **Layered Piezoelectric Effect 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 **pze11**.

#### **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 the Layered Piezoelectric Effect multiphysics coupling. The **Layered shell** and **Electric Currents in Layered Shells** lists include all applicable physics interfaces.

The default values depend on how the Layered Piezoelectric Effect 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 Layered Piezoelectric Effect 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 **Electrostatics** interface to **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 **Solid Mechanics** interface to **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 the Electrostriction multiphysics coupling. The **Solid mechanics** and **Electrostatics** lists include all applicable physics interfaces.

The default values depend on how the Electrostriction 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**, if its material type input is set to **Solid**. 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 feature 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 but keep them selected in the **Electrostatics** interface selection.

---

#### 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 **Solid model** drop down menu:

- **Isotropic** (default)
- **Cubic crystal**
- **Anisotropic**

For **Anisotropic** choice, component of a 6x6 symmetric electrostrictive coupling matrix can be entered using either **Voigt** (default) or **Standard** option for the **Material data ordering**.

For **Cubic crystal** choice, only three independent components of the electrostrictive coupling matrix need to be entered.

For **Isotropic** choice, 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 constant**  $a_1$  and **Electrostrictive constant**  $a_2$  (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 check box **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..

## Magnetostriction

---

The **Magnetostriction** multiphysics node (  ) passes the appropriate magnetization contribution from the **Magnetostrictive Material** node in the **Solid Mechanics** interface (where it is defined together with the material properties of the magnetostrictive material) to the **Ampère's Law, Magnetostrictive** node in the **Magnetic Fields** interface. It also passes the mechanics stress contribution due to the applied magnetic field back to the **Magnetostrictive Material** node.

### SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is `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 the Magnetostriction multiphysics coupling. The **Solid mechanics** and **Magnetic fields** lists include all applicable physics interfaces.

The default values depend on how the Magnetostriction 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 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 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*.

### 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.

### *Lorentz Coupling*

The **Lorentz Coupling** () a two-way multiphysics coupling feature which takes into account the interaction between the current from the **Magnetic Fields** interface and the velocity from the **Solid Mechanics** interface.

It passes the Lorentz body force  $\mathbf{F} = \mathbf{J} \times \mathbf{B}$  from the **Magnetic Fields** interface to the **Solid Mechanics** interface. It also adds the induced electric field  $\mathbf{E} = \mathbf{v} \times \mathbf{B}$  to the **Magnetic Fields** interface, where the structural velocity  $\mathbf{v}$  is defined from the displacement by the **Solid Mechanics** 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 Coupling** feature is available in 2D, 2D axisymmetric, and 3D geometries and is allowed for both time-dependent and frequency-domain studies. For

time-dependent studies, the Lorentz force contribution is accounted for only when the **Include geometric nonlinearity** check box is selected.

## SETTINGS

The **Label** is the multiphysics coupling feature name. The default **Label** (for the first multiphysics coupling feature in the model) is **Lorentz Coupling 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 **ltzc1**.

## COUPLED INTERFACES

This section defines the physics involved in the Layered Piezoelectric Effect multiphysics coupling. The **Magnetic fields** and **Solid mechanics** lists include all applicable physics interfaces.

When the Lorentz Coupling node 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.


You can also select **None** from either list to uncouple the 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*.

---

## *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 re-establish 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*.

#### **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 in this section are ignored.




See also:

- [Fluid-Structure Interaction](#) in the Structural Mechanics Modeling chapter
- [Fluid-Structure Interaction Theory](#)

---

### *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 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, 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 re-establish 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 in this section are ignored.


### *Fluid-Structure Interaction, Fixed Geometry*

---

The **Fluid-Structure Interaction, Fixed Geometry** multiphysics node could be added in versions earlier than 5.4. It has been superseded by an augmented version of the [Fluid-Structure Interaction](#) multiphysics coupling. The node will still be present and functional in models originally created in previous versions. It is, however, recommended that you update to the new way of modeling fixed geometry problems, as the **Fluid-Structure Interaction, Fixed Geometry** node may not be supported in the future.

### *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 the 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-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 re-establish 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** check box to include the forces from viscous drag along the pipe.

Select the **Centrifugal Force** check box to include centrifugal forces from the fluid in curved pipes.

Select the **Pipe flow pressure** check box 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**,  $p_o$ , — **User defined** or **Common model input**.

For **User defined**, enter a value or an expression for the external pressure.

Select the **Junction forces** check box to include local forces occurring at various connections and components in the piping system.



The check boxes in this section interact with any [Fluid Load](#) node on the same selection. When a check box 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 [The Pipe Mechanics Interface](#) section.
-


# 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](#)

## *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 <https://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** check box. 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** check box. 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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

#### *Shared Boundaries*

If the automatic selection is not sufficient, select the **Manual control of selections** check box. Then, select the connected boundaries in the **Boundary Selection** section.

The connected area is slightly extended by a distance  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

#### *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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .



- 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.

---




---

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**, or **Solid and beam parallel boundaries**.

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** check box. 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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

#### *Solid and Beam Shared Boundaries*

If the automatic selection is not sufficient, select the **Manual control of selections** check box. Then, select the connected boundaries in the **Boundary Selection** section.

The connected area is slightly extended by the distance  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. The **Automatic** option sets the value of the tolerance  $\Delta$  to 0.5% of section height of the beam. By selecting the **User defined** option, you can modify the value of the tolerance  $\Delta$ .

#### *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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

### **CONNECTION SETTINGS 3D**

Select the **Connection type** — **Solid boundaries to beam points, general**, **Solid boundaries to beam points, transition**, or **Solid boundaries to beam edges**. For the connections to beam points, there is an automatic search for possible adjacent geometrical objects. If

you want to modify the selections, select the **Manual control of selections** check box. If you clear it again, the selections will be replaced by the automatic ones.

#### *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 the automatic selection is not sufficient, select the **Manual control of selections** check box. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and select the corresponding beam points in the **Point Selection, Beam** section.

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 the automatic selection is not sufficient, select the **Manual control of selections** check box. Then, select the connected solid boundaries in the **Boundary Selection, Solid** section and select the corresponding beam points in the **Point Selection, Beam** section.

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.



- 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 <https://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** check box. 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** check box. 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** check box. 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**  $\zeta$ .
- 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.

#### *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** check box. 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.



- For more information about coupling different element types, see [Coupling Techniques](#).
- For details about the formulation of this coupling, see [Connection Between Shells and Beams](#)



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 <https://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** check box 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 check boxes 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** check box. 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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

### *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 <https://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** check box. 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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

#### *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  $\Delta$ . Select how to determine this **Connection tolerance** — **Automatic** or **User defined**. For **Automatic**, the tolerance  $\Delta$  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  $\Delta$ .

#### *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



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, Beam, or Membrane interface are embedded in solid elements. 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 <https://www.comsol.com/products/specifications/>)

## COUPLED INTERFACES

Select the **Solid** interface to couple.

Select the **Embedded structure** to couple — **Truss**, **Beam**, or **Membrane**.



When coupling the Beam interface to a Solid Mechanics interface, only the displacement degrees of freedom of the beam are constrained.

---

## CONNECTION SETTINGS

Select the **Connection type** — **Rigid**, **Spring constant per unit length**, or **Spring constant per unit surface 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 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 unit length**, the default expression for both  $k_a$  and  $k_t$  is  $1e3 * \langle \text{tag} \rangle . Eequ * \langle \text{tag} \rangle . \text{area} / h^2$ .
- For **Spring constant per unit surface area**, the default expression for both  $k_a$  and  $k_t$  is  $1e3 * \langle \text{tag} \rangle . Eequ * \langle \text{tag} \rangle . \text{perimeter} / h^2$ .

The variable  $\langle \text{tag} \rangle . Eequ$  is a placeholder for the equivalent stiffness of the Truss interface, and  $\langle \text{tag} \rangle . \text{area}$  and  $\langle \text{tag} \rangle . \text{perimeter}$  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 unit area is converted to a spring constant per unit length.



When the **Embedded structure** is a Beam interface and **Spring constant per unit length** is selected, enter the **Axial spring constant**  $k_a$ , and two **Transverse spring constants** in the local coordinate system of the beam,  $k_{y1}$  and  $k_{z1}$ . The default expression for the spring constants is  $1e3 * \langle \text{tag} \rangle . Eequ * \langle \text{tag} \rangle . \text{area} / h^2$ .

The variable  $\langle \text{tag} \rangle . Eequ$  is a placeholder for the equivalent stiffness of the Beam interface, and  $\langle \text{tag} \rangle . \text{area}$  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 unit surface 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  $1e5*\langle\text{tag}\rangle.\text{Eequ}*\langle\text{tag}\rangle.d/h^2$ .

The variable  $\langle\text{tag}\rangle.\text{Eequ}$  is a placeholder for the equivalent stiffness of the Membrane interface, and  $\langle\text{tag}\rangle.d$  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  $0[\text{unit}]*\langle\text{tag}\rangle.u_{pe}$ . The unit depends on the **Embedded structure** interface, and the variable  $\langle\text{tag}\rangle.u_{pe}$  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 be 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 box.

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** check box 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](#).
-





## 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:

$$\sigma = D\epsilon$$

**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 *visoelastic 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.





# I n d e x

- A**
  - about layered shell 1128
  - absolute values 477
  - absolute-tolerance parameters 303
  - abstract rigid domain 542
  - acoustic-structure interaction, frequency
    - domain interface 236
  - activation 230
  - activation (node) 798
    - layered shell interface 1177
  - added mass (node) 886
    - layered shell interface 1222
    - shell and plate interfaces 1103
  - added mass, interface (node) 1223
  - added mass, theory 626
  - adding
    - connections 88
  - additive strain decomposition (check box) 718, 725, 731, 740, 1253
  - adhesion (node) 939
  - alpha coefficient 776
  - Anand viscoplasticity 503
  - angle of internal friction 447, 454, 776
  - angular acceleration and velocity 538
  - anisotropic materials
    - defining 370
    - elastic properties 718, 1250
  - antisymmetry (node)
    - beam interface 1356
    - layered shell interface 1203
    - membrane 1274
    - shell and plate interfaces 1090
    - solid mechanics 853
    - truss interface 1434
  - Application Libraries window 42
  - application library examples
    - beam cross sections 1378
    - beam end release 1371
    - beams 1325
    - beam-solid connection 93
    - Cam-Clay material 728
    - circumferential mode extension 704
    - connecting shells and beams 1284
    - creep models 764
    - cross section data 1328
    - damping 810
    - decohesion 212
    - eigenfrequency analysis 56, 980
    - external stress 789
    - generation of time history from frequency domain results 62
    - geometric nonlinearity 188
    - harmonic perturbation 67
    - initial stress and strain 786, 1265
    - large plastic strains 759
    - layered shells 1146
    - linear buckling study 74
    - load cases 98, 100
    - membrane interface 1243
    - MEMS materials database 127
    - modal mass 55
    - model superposition 64
    - Mooney-Rivlin, two parameters 736
    - Ogden 736
    - one-way FSI 253
    - piezoelectric devices 1523, 1532
    - piezoelectric materials database 127
    - pipe mechanics 1463
    - plasticity models 759
    - portal crane 1287
    - prestressed bolts 74, 265
    - response spectrum analysis 75–76, 218, 285, 293

- rigid connector 905
- sharing edges 1562, 1568
- shell-beam connection 93
- shells 1028
- soil plasticity 777
- solid mechanics 707
- solid-shell connection 93
- spring foundation 234
- thermal bending of beam 1336
- thermal expansion 782
- thermal expansion at constraints 247
- thermal stress 1487
- transient contact 216
- trusses 1416
- viscoelastic material 752
- viscoplastic material 766
- applied force (node) 825, 908
  - theory 544
- applied moment (node) 828, 910
  - theory 544
- applying
  - loads 96
  - moments 101
- arbitrary Lagrangian-Eulerian (ALE)
  - method 183
- area scale factor, layered shell 1132
- area, beam cross sections 1382
- Arruda-Boyce material 429
- associated flow rule 469
- attachment (node)
  - layered shell interface 1228
  - shell and plate interfaces 1122
  - shells 1369
  - solid mechanics 915
  - theory 543
- augmented Lagrangian method 32, 197, 644
- average rotation (node) 966
- axial stress 1397
- axial symmetry
  - constraints and 109
  - initial stress and strain 356
- axisymmetric models
  - solid mechanics 79
- azimuthal mode number 348
- azimuthal wave number 631
- B**
  - back stress 465, 505
  - beam
    - coupling to a solid 90
  - beam connection (node) 1362
    - shell and plate interfaces 1112
    - solid mechanics 918
  - beam cross section interface 1399
    - theory 1382
  - beam end release (node) 1370
  - beam interface 1322
    - theory 1278, 1289
  - beams
    - cross-section data 1327
    - initial loads and strains 1296
    - initial stresses and strains 1339
    - linear elastic material 1335
    - loads applied 1361
    - prescribed acceleration 1349
    - prescribed displacement/rotation 1344
    - prescribed velocity 1347
    - section orientation 1332
    - strain-displacement/rotation 1294
    - stress evaluation 1301
    - thermal expansion 1337
    - thermal strain 1295
  - Beltrami-Michell 1388
  - bending shear stress 1397
  - bending shear stresses 1384
  - bending stress 1397
  - biaxial compression 474

- biaxial data 476
- biaxial tension 778
- bimoment 1396
- Blatz-Ko material 430
- body load (node)
  - layered shell interface 1205
  - shell and plate interfaces 1091
  - solid mechanics 856
- bolt pretension (node) 946
- bolt pre-tension, theory 260
- bolt selection (node) 947
- bolt thread contact (node) 949
- bolted joints 260
- boundary conditions
  - contact pairs 204
  - shell and plate interfaces 990
- boundary load (node) 863
  - layered shell interface 1210
- boundary nodes
  - beam interface 1325, 1464
  - layered shell 1148
  - membrane interface 1243
  - shell and plate interfaces 1028
  - solid mechanics 707
  - truss interface 1417
- boundary pair (node) 895
- box sections, beams 1328
- Bresler-Pister criterion 473
- buckling 30
- built-in couplings 88
- bulk modulus
  - elastic moduli 367, 716, 872
- Burgers vector 502
- C** calcite 477
  - calculating stress and strain 992
  - Cam-clay model 479
  - canonical systems 326
  - carbonate rocks 477
  - Cauchy stress 350
  - Cauchy stress tensor 181
  - Cayley-Hamilton theorem 338, 353
  - cell periodicity 895
  - cell periodicity (node)
    - solid mechanics 892
  - center of gravity, beam cross sections
    - 1382
  - center of mass
    - boundary (node) 830
    - edge (node) 830
    - point (node) 831
    - rigid domains 538
  - center of rotation 1386
    - boundary (node) 906
    - edge (node) 907
    - point (node) 907
    - rigid domains 540
  - ceramics 473
  - change thickness (node)
    - solid mechanics 713
  - charge conservation, piezoelectric
    - (node) 741
  - circle, Mohr 447
  - circular sections, beams 1328
  - circumferential mode number 348
  - Coble creep 502
  - coefficient of thermal expansion
    - beams 1337
    - shells and plates 1051
  - cohesion 447, 776
  - cohesionless soils 453
  - cohesive zone model 210
  - cohesive-frictional materials 473
  - common sections, beam interface 1328
  - common settings 40
  - complementarity 469
  - complex mechanical energy flux 688

- complex modulus 174
- compressive meridians 353, 447, 474–475
- concrete 473, 475
- concrete (node) 778
- conduction loss (time-harmonic) (node)
  - 814
- connection
  - beam edge to shell edge 1004
  - beam perpendicular to solid 1003
  - beam point to shell boundary 1005
  - beam point to shell edge 1006
- connections
  - beams, shells 1004
  - beams, shells, solids 997
  - shell perpendicular to solid 998
- consistency parameter 468
- constitutive relation, membranes 1239
- constraints 106
- contact (node) 921
  - layered shell interface 1188
- contact formulation 214
- continuity (node)
  - layered shell interface 1230
- continuity, layered shell 1138
- converse piezoelectric effect 141
- coordinate system, beam cross sections
  - 1384
- coordinate systems
  - constraints and 107
  - loads and 96
  - local edge system 1080
  - solid mechanics theory 326
- coordinate systems, membranes 1236
- Coulomb friction 933
- coupling
  - beam to a shell 91
  - beam to a solid 90
  - membrane to a solid 89
  - shell to a solid 89
- coupling loss (node) 813
  - layered shell interface 1185
- coupling operator 93
- crack (node) 951
- creep (node) 762
  - layered shell interface 1166
  - membrane interface 1257
  - shell and plate interfaces 1046
- creep dissipation rate density 506
- creep strain rate 494
- critical load factor 72
- cross section data (node) 1419
- cross-section data (node) 1327
- crystal cleavage 477
- crystal cut standards 144
- crystalline solids and creep 496
- curvature 1296
- cyclic symmetry, theory 630
- cylindrical coordinate systems 79

**D**

- damage (node) 794
- damped eigenfrequency study 173
- damping
  - equation of motion, and 168
  - loss factors 172
  - losses and 167
  - point mass 1362, 1437
  - solid mechanics 807
  - viscoelastic materials, and 412
- damping (node)
  - layered shell interface 1182
  - membrane interface 1270
  - shell and plate interfaces 1066
  - solid mechanics 806
- damping models 172
- decohesion (node) 941
- defining
  - anisotropic materials 370

- constraints 106
- isotropic materials 366
- multiphysics models 235
- orthotropic materials 369
- thermoelastic materials 372
- deformation gradient 330
- deformation resistance 504
  - saturation coefficient 504
  - sensitivity 504
- delamination (node) 1178
- destination point
  - boundary (node) 886
  - edge (node) 886
  - point (node) 886
- destination point (node) 885
- destinations and sources 201
- deviatoric creep 498, 501
- deviatoric stress 441
- dielectric loss (node) 813
  - layered shell interface 1186
- dielectric loss factor 153
- diffusional creep 502
- dilatational contributions 417
- direct piezoelectric effect 141, 513
- discretization, layered shell 1133
- dislocation creep 503
- dispersion curves 58
- displacement field, defining 177
- displacement gradients 329
- displacement variables
  - element types and 87
- dissipated energy 687
- dissipated energy density 472, 506
- dissipated energy density rate 394
- dissipation, piezoelectric materials 519
- distortional contributions 417
- disturbance factor 478
- documentation 40
- dolomite 477
- domain nodes
  - solid mechanics 707
- double dogleg nonlinear solver 220
- Drucker-Prager criterion 448
- ductile materials 444
- Dulong-Petit law 372
- dynamic cyclic symmetry 630
- dynamic frictional coefficients 653
- E** edge group (node) 1371
- edge load (node)
  - beam interface 1358
  - layered shell interface 1212
  - shell and plate interfaces 1094
  - solid mechanics 865
  - truss interface 1436
- edge nodes
  - beam interface 1325, 1464
  - membrane interface 1243
  - solid mechanics 707
  - truss interface 1417
- effective stress 1398
- effective stress tensor 478
- eigenfrequency study 52
  - solid mechanics 579
- eigenvalue solvers 53
- eight-chain model 429
- elastic deformation tensor 340
- elastic energy 685
- elastic Green-Lagrange strain tensor 341
- elastic material properties 716–718, 1249–1250
- elastic moduli 366
- elastic predeformation (node) 960
- elastic right Cauchy-Green tensor 340
- elastic strain energy 435
- elasticity matrix 366
- elastoplastic materials 439

- electrostriction (node) 1545
- electrostriction multiphysics interface
  - 1528
- element types 87
- elliptic cap 451
- elplastic 469–470
- emailing COMSOL 42
- embedded reinforcement (node) 1571
- energy dissipation 687
- energy function 415
- energy quantities 685
- equation of motion, damping and 168
- equivalent creep strain rate 498
- equivalent plastic strain rate 440
- equivalent single layer shell theory 1008
- equivalent viscous damping 173
- evanescent modes 58
- excitation frequency 173
- explicit damping 174
- external loads, shell and plate interfaces
  - 991
- external strain (node) 790, 793
  - layered shell interface 1174
- external stress (node) 787, 1340, 1427
  - layered shell interface 1173
  - membrane interface 1266
  - shell and plate interfaces 1058
- external stress-strain relation (node) 815
- F** face load (node) 953
  - layered shell interface 1207
  - shell and plate interfaces 1092
- failure surfaces 439
- ferroelectroelasticity multiphysics inter-
  - face 1529
- Filter 232
- first order shear deformation laminate
  - theory 1010
- first Piola-Kirchhoff stress 350
- fixed constraint (node) 821, 843
  - layered shell interface 1197
  - shell and plate interfaces 1070
- fixed constraint, interface (node) 1198
- fixed constraint, theory 543
- Floquet periodicity, theory 630
- flow rule 468
- fluid and pipe materials (node) 1472
- fluid pore pressure 453
- fluid-membrane interaction, fixed geom-
  - etry 1510
- fluid-pipe interaction 1556
- fluid-pipe interaction, fixed geometry
  - 1512
- fluid-shell interaction 1504
- fluid-solid interaction 1499, 1508, 1513
- fluid-structure interaction 1502, 1551,
  - 1556
- fluid-structure interaction interface
  - theory 636
- fluid-structure interaction, fixed geom-
  - etry 1506
- fluid-structure interaction, pair 1554
- fold lines 988
- foldline connection, layered shell 1137
- fold-line limit angle 1025, 1144
- follower loads 1239
- force linear strains (check box) 718, 724,
  - 730, 740, 743, 746, 816, 1161, 1250
- free (node) 833
  - layered shell interface 1189
- free-free modes 55
- frequency domain study
  - solid mechanics 578
- frequency domain, prestressed study 188
- frequency response study
  - loss factor damping 173
- friction (node) 932

- friction forces 935
- friction models 933
- friction, angle 447, 454, 776
- frictionless materials 448, 450
- G**
  - Gao material 431
  - Garofalo law 501
  - Gauss points 469–470
  - general extrusion operator 93
  - generalized Hoek-Brown criterion 477
  - Generalized Kelvin–Voigt viscoelastic model 395
  - generalized Maxwell model 390
  - Gent material 430
  - geological strength index (GSI) 477
  - geometric nonlinearity 184
    - membranes 1234
    - micromechanics, and 176
    - piezoelectric devices 189
  - glass transition temperature 410
  - global
    - coordinate systems 326
  - GMRES iterative solvers 302
  - gradient displacements 329
  - gravity (node) 858
    - layered shell interface 1208
    - pipe mechanics interface 1478
  - Green’s theorem 1391
  - Green-Lagrange strain 177
  - Green-Lagrange tensor 332
- H**
  - Haigh–Westergaard coordinates 354
  - hardening constant 504
  - hardening models
    - theory 460
  - hardening sensitivity 504
  - harmonic loads 578
  - harmonic perturbation 65
  - harmonic perturbation (node)
    - shell and plate interfaces 1125
    - truss interface 963, 1373
- hear center 1386
- hear correction factor 1385
- heat dissipation 174
- Hencky plastic strain 471
- Hencky strain 335
- Hermitian matrices 301
- hexagonal prism 442
- Hill orthotropic plasticity 444
- Hill’s effective stress 446
- Hoek-Brown criterion 476
- hole (node) 1400
- H-profile sections, beams 1328
- hydrostatic axis 354
- hydrostatic pressure 351, 446
- hydrostatic stress 453
- hygroscopic swelling 103
  - hygroscopic swelling (node) 783
    - layered shell interface 1170
    - membrane interface 1262
  - hygroscopic swelling (node), multiphysics 1540
- hyperbolic sinus 501
- hyperelastic material 686
- hyperelastic material (node) 731
  - layered shell interface 1156
- hyperelastic materials 422
  - nearly incompressible 419
  - theory 415
- hysteretic loss 150
- I**
  - I.R.E. standard, for material orientation 142
  - ideal plasticity 460
  - IEEE standard, for material orientation 142
  - imperfection sensitivity 73
  - implementation
    - beams 1297

- trusses 1408
- include geometric nonlinearity (check box) 719
- inelastic deformation tensor 341
- inertial effects, contact modeling 214
- inertial forces 539
- initial loads and strains, beams 1296
- initial stress and strain 520
  - theory 355
- initial stress and strain (node) 785
  - beam interface 1339
  - layered shell interface 1171
  - membrane interface 1264
  - shell and plate interfaces 1055
  - solid mechanics 785
  - truss interface 1427
- initial values (node)
  - beam interface 1327
  - layered shell interface 1153
  - rigid domains 820
  - rigid domains, theory 540
  - shell and plate interfaces 1031
  - solid mechanics 712
- initial yield stress 461
- integration in a laminate 1010
- internet resources 40
- invariants 415
- inverse piezoelectric effect 514
- isochoric
  - contributions 417
  - process 330
  - strain energy density 419
- isochoric strain energy 436
- isochoric-elastic
  - deformation 418
  - Green-Lagrange strain tensor 418
  - right Cauchy-Green tensor 418
- isotropic hardening 461
- isotropic materials
  - defining 366
  - elastic properties 716, 1249
- isotropic plasticity 439
- isotropic rocks 477
- iterative solvers 301

**J**

- j-integral (node) 954
- joints, bolted 260
- Joule heating and thermal expansion interface 236, 1494

**K**

- k coefficient 777
- Kelvin–Voigt viscoelastic model 399
- kinematic constraints 115
- kinematic hardening 464
- kinematics, rigid domain 536
- kinetic energy 687
- knowledge base, COMSOL 43
- Kuhn-Tucker conditions 469

**L**

- Lade-Duncan criterion 453
- Lagrange elements 84
- Lagrange shape functions, trusses and 1406
- Lagrangian formulations 328
- Lamé parameters 367, 716
- lamina constitutive law 1011
- laminate constitutive law 1014
- large deformation modeling 328
- large deformations 31
  - piezoelectric materials 189
- large plastic strain 468
- large strain plasticity 417
- layered linear elastic material (node)
  - membrane interface 1251
  - shell and plate interfaces 1036
- layered shell interface 1143
  - theory 1128
- left Cauchy-Green tensor 333



- Lie derivative 469
- limestone 477
- limiting chain extensibility 430
- line load (node) 1213
- linear buckling study 30, 188
- linear elastic material (node) 713
  - beam interface 1335
  - layered shell interface 1154
  - membrane interface 1247
  - shell and plate interfaces 1034, 1039
  - truss interface 1420
- linear elastic materials 123
- linear viscoelasticity 388
- linearized buckling analysis 72
- linper operator 65
- load cases 98
- load multiplier 72
- loads
  - applied to beams 1361
  - pressure 102
  - singular 100
  - total 103
- local coordinate systems 327
- local edge system 1080
- location
  - boundary (node) 827
  - edge (node) 827
  - point (node) 827
- locking 421
- Lode angle 353
  - Hoek-Brown criterion 477
  - Mohr-Coulomb criterion 447
  - Tresca criterion 442
  - Willam-Warnke criterion 474
- logarithmic decrement 168
- logarithmic plastic strain 471
- long-term shear modulus 391
- loss factor damping
  - solid mechanics and 172
  - solid mechanics theory 172
  - springs, and 234
- loss modulus 174, 412
- losses and damping 167
- low-reflecting boundary (node) 896
- low-reflecting boundary, theory 629

**M**

- macroscopic shear modulus 429
- magnetostriction 521
- magnetostriction (node) 1549
- magnetostriction multiphysics interface 1530
- magnetostrictive material 744
- magnetostrictive material (node) 744
- marble 477
- mass and moment of inertia (node) 829, 911
  - theory 542
- mass density 330
- mass matrix scaling 69
- mass moment of inertia 1361
- material coordinates 325
- material frame 177
- material models 120
- materials
  - nearly incompressible 124, 375
  - piezoelectric 514
  - viscoelastic 388
- materials, hyperelastic 415
- Matsuoka-Nakai criterion 452
- max scaling 69
- max shear stress factor 1385
- mechanical damping (node) 811
  - layered shell interface 1184
- mechanical energy flux 688
- membrane
  - coupling to a solid 89
- membrane interface 1241

- theory 1234
- meridians, tensile and compressive 353, 447, 474–475
- metal plasticity 440
- metals 444, 449
- metals and creep 496
- Mindlin plate theory 998
- Mindlin-Reissner type shell 977
- MITC shell formulation 977, 987
- mixed formulations 124, 375
- mobilized planes 452
- mode analysis study 57–58
- modeling, large deformations 328
- modified Cam-clay model 479
- modified Mohr-Coulomb criterion 455
- modified tensors 418
- Mohr's circle 1383
- Mohr-Coulomb criterion 446
- moment computations 116
- moments 539
  - beams 1331, 1468
  - shells and plates 991
  - solid mechanics and 101
- moments of inertia 1278, 1376
- moments of inertia, beam cross sections 1382
- Mooney-Rivlin material
  - five parameters 423
  - nine parameters 424
  - two parameters 423
- moving mesh interface, piezoelectric devices and 191
- MPH-files 42
- mu coefficient 777
- Mukherjee-Bird-Dorn equation 503
- mullins effect (node) 752
  - layered shell interface 1162
- multiaxial stress states 473
- multibody dynamics interface 915
- multiphysics
  - electrostriction 1528, 1545
  - ferroelectroelasticity 1529
  - hygroscopic swelling 1540
  - magnetostriction 1530, 1549
  - piezoelectric effect 1542
  - piezoelectricity 1521
- multiphysics coupling
  - fluid-membrane interaction, fixed geometry 1510
  - fluid-pipe interaction 1556
  - fluid-pipe interaction, fixed geometry 1512
  - fluid-shell interaction 1504
  - fluid-solid interaction 1499, 1508, 1513
  - fluid-structure interaction 1502, 1551, 1556
  - fluid-structure interaction, fixed geometry 1506
  - fluid-structure interaction, pair 1554
  - Joule heating and thermal expansion 1494
  - thermal expansion (node) 1534
  - thermal stress 1486
- multiphysics modeling 235
- MUMPS direct solvers 301
- Murnaghan material 432

**N**

- Navarro-Herring creep 502
- Navier-Stokes equations 636
- nearly incompressible hyperelastic materials 419
- nearly incompressible materials 124, 356, 375
- Neo-Hookean material 422
- Neumann boundary conditions
  - applied force 825
  - applied moment 828

- no rotation (node)
  - beam interface 1353
  - shell and plate interfaces 1083
- nodes, common settings 40
- nominal stress 359
- nonassociated flow rule 469
- nonlinear elastic material 686
- nonlinear elastic material (node) 720
- Norton equation 498
- Norton-Bailey model 499
- O**
  - octahedral plane 354
  - Ogden material 426
  - one-way model formulations 637
  - orientation, piezoelectric material 142
  - orthotropic materials
    - defining 369
    - elastic properties 717, 1249
    - loss factor damping, and 809
  - Ottosen criterion 475
  - over-consolidation pressure 483
  - overpressure 1441
- P**
  - pair nodes
    - beam interface 1325, 1464
    - membrane interface 1243
    - solid mechanics 707
    - truss interface 1417
  - parametric analysis 30
  - PARDISO direct solvers 301
  - penalty factors
    - contact node, and 926
    - contact pairs and 205
  - penalty method 644
  - perfectly elastoplastic materials 460
  - perfectly plastic hardening 460
  - periodic boundary conditions 889, 892
  - periodic condition (node)
    - shell and plate interfaces 1106
    - solid mechanics 889
  - periodic conditions, theory 630
  - phase (node) 962
    - beam interface 1372
    - layered shell interface 1215
    - shell and plate interfaces 1124
    - solid mechanics 962
    - truss interface 1438
  - physics interfaces
    - connecting to DOF 87
  - physics interfaces, common settings 40
  - piezoelectric coupling 1521
  - piezoelectric crystal cut 142
  - piezoelectric effect (node) 1542
  - piezoelectric losses 150
  - piezoelectric material (node) 741
    - layered shell 1158
  - piezoelectricity multiphysics interface
    - 1521
  - piezomagnetic coupling 745
  - piezomagnetic coupling matrices 522
  - pinned (node)
    - beam interface 1351
    - shell and plate interfaces 1081
    - truss interface 1430
  - pipe cross section (node) 1465
  - pipe mechanics interface 1461
    - theory 1440
  - pipe sections, beams 1328
  - pipes
    - stress evaluation 1449
  - plane stress and strain 78, 703
  - planes, symmetry and constraints 110
  - plastic deformation gradient 342
  - plastic element 469–470
  - plastic flow rule 471
  - plastic Green-Lagrange strain 342
  - plastic multiplier 468
  - plastic potential 468

- plasticity (node) 753
  - layered shell interface 1163
  - membrane interface 1255
  - shell and plate interfaces 1044
- plasticity models 754
- plate interface 1024
  - theory 983
- plates
  - external loads 991
  - initial stresses and strains 1055
  - linear elastic material 1034
  - MITC shell formulation 987
  - prescribed acceleration 1079
  - prescribed displacement/rotation 1071
  - prescribed velocity 1076
  - stress and strain calculations 992
  - symmetry and antisymmetry 990
- point load (node) 866
  - beam interface 1360
  - layered shell interface 1214
  - shell and plate interfaces 1096
- point load on axis (node) 868
- point loads example 100
- point mass (node)
  - beam interface 1361
  - shell and plate interfaces 1105
  - truss interface 1437
- point mass damping (node)
  - beam interface 1362
  - shell and plate interfaces 1106
  - truss interface 1437
- point nodes
  - beam interface 1325, 1464
  - membrane interface 1243
  - solid mechanics 707
  - truss interface 1417
- Poisson type equation 1388
- Poisson's ratio 124, 367, 375, 716, 872
- polynomial hyperelastic material 424
- port (node) 956
- power law 498
- Prager's hardening rule 465
- Prandtl stress function 1393
- predeformation (node) 879
  - layered shell interface 1221
  - shell and plate interfaces 1103
- prescribed acceleration (node) 840
  - beam interface 1349
  - layered shell interface 1194
  - shell and plate interfaces 1079
- prescribed acceleration, interface (node)
  - 1196
- prescribed displacement (node)
  - layered shell interface 1190
  - solid mechanics 835
- prescribed displacement, interface
  - (node) 1191
- prescribed displacement, theory 543
- prescribed displacement/rotation (node)
  - 822
  - beam interface 1344
  - shell and plate interfaces 1071
  - theory 543
- prescribed velocity (node) 838
  - beam interface 1347
  - layered shell interface 1192
  - shell and plate interfaces 1076
  - solid mechanics 838
- prescribed velocity, interface (node)
  - 1193
- pressure loads 102
- pressure-wave speeds 367, 717
- prestressed analysis, eigenfrequency
  - study 188
- primary creep 495
- principal stresses 352, 354

- principal stretches 415
- principle of virtual work 576
- Prony series 390
- propagating modes 58
- Q** quaternion constraint 537
- quaternion representation, of rigid connector 641
- R** radius of gyration 1383
- random vibration analysis 30
- rate independent plasticity 468
- rate of strain tensor 338, 470
- Rayleigh damping 170
- rectangle sections, beams 1328
- reference coordinates 325
- reference point for moment computation 116
- refpnt variable 705
- renaming
  - displacement DOF 88
- representative volume element 257, 638
- resonant frequency 170
- response spectrum analysis 30
- results evaluation in a laminate 1021
- results evaluation, for shells 995
- results evaluation, layered shells 1141
- right Cauchy-Green deformation tensor 332, 415
- right polar decomposition 331
- right stretch tensor 331
- rigid body 535
- rigid connection type, shells 999
- rigid connector 641
- rigid connector (node) 898, 1364
  - layered shell interface 1224
  - shell and plate interfaces 1114
- rigid connector theory 543
- rigid connector, interface (node) 1226
- rigid connectors
  - kinematic constraints and 115
  - moments and 101
- rigid domain (node) 817
  - layered shell interface 1187
  - pipe mechanics interface 1474
  - theory 535
- rigid domains
  - angular acceleration 539
  - angular velocity 538
  - dynamics 539
- rigid motion suppression (node) 854
  - layered shell interface 1204
- ring load (node) 867
  - shell and plate interface 1098
- rock mass 478
- rock types 477
- rocks 476
- rocks (node) 779
- roller (node) 847
  - layered shell interface 1200
- roller, interface (node) 1201
- rotated coordinate system 145
- rotating frame (node) 860
  - layered shell interface 1209
- rotation matrix 537
- rotation, rigid body 536
- rotational degrees of freedom 1279
- rotational joints, beams 115
- S** safety (node) 557, 800, 1401
  - layered shell interface 1181
  - membrane interface 1268
  - shell and plate interfaces 1064
- Saint-Venant torsion 1397
- saturated cohesive soils 444
- scaling of eigenvectors 53
- second Piola-Kirchhoff stress 181, 327, 350
- secondary creep 495

- section orientation (node) 1332
  - pipe mechanics interface 1470
- section stiffness (node)
  - shell and plate interfaces 1068
- selecting
  - solvers 300
- serendipity elements 84
- shape factors 476
- shape memory alloy 508
- shear area 1385
- shear correction factor computation
  - 1019
- shear modulus expression 367, 716, 872
- shear stresses 352, 447
- shear-wave speeds 367, 717
- shell
  - coupling to a beam 91
  - coupling to a solid 89
- shell connection (node)
  - beams 1364
  - solid mechanics 919
- shell interface 1024
  - theory 983
- shells
  - external loads 991
  - initial stresses and strains 1055
  - linear elastic material 1034
  - MITC shell formulation 987
  - prescribed acceleration 1079
  - prescribed displacement/rotation 1071
  - prescribed velocity 1076
  - stress and strain calculations 992
  - symmetry and antisymmetry 990
- shift function 747
- sign conventions 351
- simply supported (node) 1084
- singular loads 100
- size factors 476
- skew-symmetric part 338
- slip velocity (node) 936
- slit boundary 260
- SLS model 400
- small plastic strain 467
- soil deformation theory 484
- soil plasticity (node) 775
- solid
  - coupling to a beam 90
  - coupling to a membrane 89
  - coupling to a shell 89
- solid connection (node) 1111
- solid mechanics
  - damping 807
  - edge loads 865
  - initial stresses and strains 785
  - prescribed acceleration 840
  - prescribed velocity 838
- solid mechanics interface 702
  - theory 323
- solid\_epdm 960
- solid-shell connection (node) 1559, 1562, 1566
- solver methods
  - augmented Lagrangian 644
  - penalty method 644
- solver parameters 300
- solver settings 300
- SOR line solvers 302
- source point
  - boundary (node) 886
  - edge (node) 886
  - point (node) 886
- source point (node) 885
- sources and destinations 201
- spatial coordinates 325
- spatial stress tensor 181
- spatially mobilized planes (STP) 453

- spin tensor 338
- spring constant 233
- spring foundation 831
  - spring foundation 831, 869, 912
    - layered shell interface 1216
    - shell and plate interfaces 1099
  - spring foundation, interface (node) 1217
  - spring foundation, solid mechanics 233
  - spring foundation, theory 616
  - spring-damper (node) 880
  - spring-damper material (node) 1429
- St Venant torsion 1395
- St Venant-Kirchhoff material 422
- St. Venant's principle 100
- standard linear solid model 400
- standard settings 40
- static frictional coefficients 653
- stationary solvers 300
- steady-state creep 495
- steady-state stiffness 391
- storage modulus 174, 412
- Storakers material 427
- stored energy 686
- straight edge constraint (node) 1420
- straight edges 1409
- strain energy density 436, 686
- strain-displacement, trusses 1406
- strain-displacement/rotation 1294
- strains, membranes 1236
- stress
  - Cachy 350
  - first Piola-Kirchhoff 350
  - second Piola-Kirchhoff 350
- stress and strain, piezoelectric devices
  - 166, 514, 522
- stress components, beam cross sections
  - 1387
- stress evaluation, beams 1301
- stress evaluation, pipes 1449
- stress linearization (node) 965
- stress stiffening 1234
- stresses, membranes 1236
- stress-strain relation
  - beams 1294
  - trusses 1407
- structure-pipe connection (node) 1570
- study steps, geometric nonlinearity and
  - 184
- study types
  - eigenfrequency 579
  - frequency domain, solid mechanics 578
  - parametric 30
- surface traction and reaction forces 118
- symmetric matrices 300
- symmetry (node)
  - beam interface 1354
  - layered shell interface 1202
  - membrane interface 1272
  - shell and plate interfaces 1085
  - solid mechanics 849
  - truss interface 1432
- symmetry constraints 109
- symmetry plane (node) 1088
  - solid mechanics 851

**T**

- tangent modulus 461
- tangential strains 1238
- technical support, COMSOL 42
- temperature loads 102
- tensile meridian 353
- tensile meridians 447, 449, 454, 474-475
- tension cut-off 454
- tertiary creep 495
- theory
  - beam cross sections 1382
  - beam interface 1278, 1289
  - fluid-structure interaction 636

- layered shell interface 1128
- membrane interface 1234
- pipe mechanics interface 1440
- shell and plate interfaces 983
- solid mechanics interface 323
- theory background of laminated shell
  - 1006
- thermal expansion
  - loads and 102
- thermal expansion (for constraints)
  - (node)
    - layered shell interface 1199
    - shell and plate interfaces 1075
- thermal expansion (multiphysics coupling) 1534
- thermal expansion (node) 780
  - beam interface 1337
  - layered shell interface 1168
  - membrane interface 1260
  - pipe mechanics interface 1476
  - shell and plate interface 1049
  - solid mechanics 780, 845, 1352, 1431
- thermal expansion, hyperelastic materials and 416
- thermal strain
  - beams 1295
- thermal stress interface 1486
  - modeling 235
- thermal-electric-structural interaction
  - 236
- thermoelastic materials, defining 372
- thermorheologically simple viscoelastic materials 409
- thickness and offset (node)
  - membrane interface 1246
  - shell and plate interfaces 1032
- thin elastic layer (node) 875
  - layered shell interface 1219
  - thin elastic layer, interface (node) 1220
  - thin elastic layer, solid mechanics 233
  - thin elastic layer, theory 618
  - thin-film damping (node) 897
  - thread boundary selection (node) 950
  - three-parameter model 400
  - torsion, beam cross sections 1393
  - torsional constants and moments 1298
  - torsional shear stress 1397
  - total Lagrangian formulation 31
  - total loads 103
  - t-profile sections, beams 1328
  - tractions 181
  - transient creep 495
  - translation, rigid body 536
  - Tresca effective stress variable 443
  - Tresca stress 441
  - Tresca yield criterion 442
  - triaxial conditions 473
  - triaxial data 476
  - TRS material 410
  - true strain 335
  - true stress tensor 181
  - truss interface 1415
  - trusses
    - harmonic perturbation 963, 1373
    - initial stresses and strains 1427
    - linear elastic material 1420
    - straight edge 1409
    - strain-displacement 1406
  - two-point tensor 330

**U**

  - undrained shear strength 442
  - uniaxial compression 473, 475, 477, 565
  - uniaxial tension 441, 778
  - units, loads and 96
  - u-profile sections, beams 1328
  - user defined material 434
  - using



- coordinate systems 326
- predefined variables 116
- spatial and material coordinates 325
- weak constraints 117
- uspring variable 233
- V** Varga material 428
- variables
  - cross section data, beams 1378
  - deformation gradient tensor 331
  - density 331
  - elastic Green-Lagrange tensor 341
  - elastic right Cauchy-Green tensor 341
  - elastic, inelastic, and total volume ratio 341
  - equivalent creep strain 498
  - equivalent plastic strain 440, 464
  - Green-Lagrange tensor 333
  - invariants 435
  - isochoric elastic Green-Lagrange strain tensor 419
  - isochoric-elastic Cauchy-Green deformation tensor 418
  - isochoric-elastic right Cauchy-Green deformation tensor 419
  - material and spatial coordinates 326
  - predefined 116
  - principal elastic stretches 435
  - refpnt 705
  - right Cauchy-Green deformation tensor 333
  - right stretch tensor 331
  - rotation tensor 331
  - stresses, beams 1380
  - thermal stretch and thermal volume ratio 417
  - Tresca effective stress 443
  - volumetric plastic strain 440, 480, 484
- vdamper variable 234
- Vicat softening temperature 411
- Villari effect 521
- viscoelastic materials
  - definition 123
  - frequency domain analysis and damping 412
  - temperature effects 409
  - theory 388
- viscoelasticity (node) 746
  - layered shell interface 1161
  - membrane interface 1254
  - shell and plate interfaces 1043
- viscoplasticity (node) 765, 771
  - layered shell interface 1167
  - membrane interface 1259
  - shell and plate interfaces 1047
- viscous damping 174, 234
- Voigt form 143
- Voigt notation 368, 809
- volume reference density 240
- volume-preserving contributions 417
- volumetric contributions 417
- volumetric creep strain rate 497
- volumetric plastic strain rate 440
- volumetric plastic strain variable 480, 484
- volumetric strain energy 435
- volumetric strain energy density 419
- von Mises criterion 440–441
- von Mises effective stress 1398
- von Mises stress 441
- W** warping function 1393
- warping, beam cross sections 1395
- wave speeds 717, 1249, 1336
- wave speeds (node) 969
- waveguide 58
- weak constraints, using 117
- wear (node) 943
- websites, COMSOL 43

Weertman creep 503  
Willam-Warnke criterion 473  
WLF shift functions 410, 747

- Y** yield function 440  
yield functions 447  
yield stress levels 755, 767  
yield surface 469  
yield surfaces 439, 448  
Young's modulus expression 367, 716,  
872
- Z** Zener model 400