

Structural Mechanics Module

User's Guide

Structural Mechanics Module User's Guide

© 1998–2018 COMSOL

Protected by patents listed on www.comsol.com/patents, and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,457,932; 8,954,302; 9,098,106; 9,146,652; 9,323,503; 9,372,673; and 9,454,625. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement (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 Server, and LiveLink are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see www.comsol.com/trademarks.

Version: COMSOL 5.4

Contact Information

Visit the Contact COMSOL page at 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 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. Other useful links include:

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

Part number: CM021101

C o n t e n t s

Chapter 1: Introduction

About the Structural Mechanics Module	22
Why Structural Mechanics is Important for Modeling	22
What Problems Can It Solve?	23
The Structural Mechanics Module Physics Interface Guide	26
Common Physics Interface and Feature Settings and Nodes	30
Geometry Levels for Study Capabilities	30
Where Do I Access the Documentation and Application Libraries?	32
Overview of the User's Guide	36

Chapter 2: Structural Mechanics Modeling

Study Types	41
Introduction	41
Stationary Analysis	41
Eigenfrequency Analysis	42
Mode Analysis	46
Time Domain Analysis	47
Frequency Domain Analysis	48
Modal Superposition	51
Harmonic Perturbation	53
Modal Reduced Order	56
Linearized Buckling Analysis	61
Bolt Pretension Study	62
Response Spectrum Analysis Study	63
Selecting the Physics Interface	65
Solid Mechanics	65
Shell and Plate	68
Membrane	68

Beam	69
Truss	69
Selecting Discretization	70
Shape Function Order	70
Lagrange and Serendipity Shape Functions	71
Choosing Shape Functions in Multiphysics Models	71
Implicit Shape Function Orders	72
Coupling Different Element Types	74
Introduction to the Element Types	74
Coupling Techniques	74
Applying Loads	81
Units, Orientation, and Visualization	81
Load Cases	83
Singular Loads	84
Moments in the Solid Mechanics Interface	85
Pressure	86
Acceleration Loads	86
Temperature Loads—Thermal Expansion	86
Hygroscopic Swelling	87
Total Loads	87
Defining Constraints	89
Rigid Body Motion	89
Orientation	90
Prescribed Displacements, Velocities, and Accelerations	91
Symmetry Constraints	92
Elemental and Nodal Constraints	94
Suppressing Constraints on Lower Dimensions	96
Kinematic Constraints	97
Rotational Joints	98
Calculating Reaction Forces	99
Using Predefined Variables to Evaluate Reaction Forces	99
Using Weak Constraints to Evaluate Reaction Forces	100
Using Surface Traction to Evaluate Reaction Forces	101

Evaluating Surface Traction Forces on Internal Boundaries	102
Introduction to Material Models	103
Material Models for Structural Mechanics	103
Combination of Material Models	104
Entering material data	105
Introduction to Linear Elastic Materials	105
Introduction to Viscoelastic Materials	106
Mixed Formulation	106
Modeling Damage	107
About the Material Databases for the Structural Mechanics Module	108
Using External Materials	109
Modeling Piezoelectric Problems	122
About Piezoelectric Materials	122
Piezoelectric Material Orientation	123
Piezoelectric Losses.	131
References for Piezoelectric Damping	136
Piezoelectric Coupling	137
Create the Piezoelectric Effect Interface and Define Domains	137
Complete Settings of Piezoelectric Materials	139
Add Damping and Loss	140
Define Material Properties	140
Coupling Piezoelectric Devices with Acoustics	141
Modeling Magnetostrictive Materials	143
Magnetostriction Coupling	143
Create the Magnetostriction Interface and Define Domains.	143
Linear vs Nonlinear Magnetostriction	145
Mechanical Damping and Losses	146
Damping Sources.	146
About Damping	147
Rayleigh Damping	149
Loss Factor Damping	151
Viscous Damping	152
Equivalence Between Loss Factor, Rayleigh, and Viscous Damping	152
Explicit Complex Valued Damping	153

Viscoelastic Damping	153
Piezoelectric Damping	153
Adding Damping in the Modal Solver	153
Modeling Geometric Nonlinearity	155
Geometric Nonlinearity, Frames, and the ALE Method	155
Geometric Nonlinearity for the Solid Mechanics Interface	163
Geometric Nonlinearity for the Shell, Plate, Membrane, Beam and Truss Interfaces	164
Solving Geometrically Nonlinear Problems	164
Prestressed Structures.	165
Geometric Nonlinearity for the Piezoelectric Material.	167
References	170
Contact Modeling	171
General	171
Setting up a contact problem	171
Contact Pairs	174
Meshing for Contact Analysis	175
Constraints	176
Settings for Contact Nodes	177
Time-Dependent Contact Analysis	182
Multiphysics Contact	182
Fallback Nodes for Contact	183
Solver Settings for Contact Analysis.	183
Monitoring the Solution	185
Dependent Variables in Contact Analysis.	185
Important Contact Variables	188
Activating and Deactivating Material	190
Springs and Dampers	193
Defining Multiphysics Models	195
Thermal-Structure Interaction.	195
Acoustic-Structure Interaction.	196
Thermal-Electric-Structural Interaction	196

Thermally Coupled Problems	198
Temperature-Dependent Material Data	198
Thermal Expansion	200
Constraints and Thermal Expansion.	206
Thermoelastic Damping	207
Fluid-Structure Interaction	209
Deforming Fluid Domains	209
Union or Assembly	210
Contact and FSI	211
Computing Mass Properties	212
Effective Properties of Periodic Structures	215
RVE Modeling Using the Cell Periodicity Node	215
Pretensioned Bolts	217
Simplified Modeling of Bolt Threads	223
Performing a Response Spectrum Analysis	225
Stress Linearization	233
Solver Settings	239
Symmetric Matrices.	239
Selecting Iterative Solvers	240
Solver Settings for Viscoelasticity, Creep, and Viscoplasticity	241
Specifying Tolerances and Scaling for the Solution Components	242
Solver Settings.	242

Chapter 3: Structural Mechanics Theory

Solid Mechanics Theory	248
Introduction	248
Frames and Coordinate Systems	250
Material and Spatial Coordinates	250
Coordinate Systems.	251
Analysis of Deformation	253
Lagrangian Formulation	253
Deformation Measures	256
Invariants of Strain	259
Inelastic Strain Contributions	263
Axial Symmetry and Deformation	266
Generalized Plane Strain	269
Out-of-plane and Circumferential Modes.	271
Stresses	274
Defining Stress.	274
Invariants of the Stress Tensor	275
Plane Strain and Plane Stress Cases	278
Initial Stresses and Strains	279
External Stress.	280
Axial Symmetry and Stresses	281
Equation of Motion	282
Material Models	285
Linear Elastic Material	288
Mixed Formulation	296
Nonlinear Elastic Materials	298
Linear Viscoelastic Materials	308
Large Strain Viscoelasticity	317

Hyperelastic Material Models	319
Elastoplastic Material Models	340
Failure Criteria for Concrete, Rocks, and Other Brittle Material	372
Elastoplastic Soil Models	377
Creep and Viscoplasticity.	389
Shape Memory Alloy	403
Piezoelectric Material	407
Magnetostrictive Material.	412
Rigid Domain Model	419
Damage Models	429
Safety Factor Evaluation	435
Formulation of the Equilibrium Equations	446
Equation Implementation	446
Study types	448
Frequency-Domain Studies	448
Eigenfrequency Studies	449
Response Spectrum Analysis Theory	452
Linear Buckling	470
Damping	472
Rayleigh Damping Theory	472
Loss Factor Damping Theory	473
Viscous Damping Theory.	474
Loads and Boundary Conditions	476
Distributed Loads	476
Rotating Frame and Gravity	477
Spring Foundation and Thin Elastic Layer	479
Added Mass.	483
Rigid Motion Suppression Boundary Condition	484
Low-Reflecting Boundary Condition	487
Cyclic Symmetry and Floquet Periodic Conditions	488
Symmetry Condition with Translation	489
Roller Boundary Condition	490
Thermal Expansion of Constraints	492
Fluid-Structure Interaction Theory	494

Periodic Cell Theory	496
Rigid Connector Theory	497
Contact Analysis Theory	498
Stress Linearization	510
Energy Quantities	513
Elastic Energy	513
Kinetic Energy	515
Dissipated Energy	515
Mechanical Energy Flux	516
Energy Variables	517
References	520

Chapter 4: Solid Mechanics

The Solid Mechanics Interface	530
Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics	534
Initial Values	538
Change Thickness	539
Linear Elastic Material	539
Nonlinear Elastic Material	546
Elastoplastic Soil Material.	552
Hyperelastic Material	556
Shape Memory Alloy	561
Piezoelectric Material	565
Magnetostrictive Material.	568
Viscoelasticity	570
Mullins Effect	574
Plasticity	575
Creep	579
Viscoplasticity	583
Porous Plasticity	588
Soil Plasticity	590
Concrete.	592

Rocks	593
Thermal Expansion (for Materials)	594
Hygroscopic Swelling	597
Initial Stress and Strain.	598
External Stress.	600
External Strain.	602
Damage	605
Activation	607
Safety	608
Damping	613
Mechanical Damping	617
Coupling Loss	618
Dielectric Loss.	619
Conduction Loss (Time-Harmonic)	620
External Stress-Strain Relation.	621
Rigid Domain	622
Initial Values (Rigid Domain)	625
Fixed Constraint (Rigid Domain)	627
Prescribed Displacement/Rotation	627
Applied Force (Rigid Domain)	630
Location Nodes	632
Applied Moment (Rigid Domain)	632
Mass and Moment of Inertia (Rigid Domain)	633
Center of Mass Nodes.	634
Spring Foundation (Rigid Domain)	635
Free.	637
Prescribed Displacement.	638
Prescribed Velocity	642
Prescribed Acceleration	644
Fixed Constraint	646
Thermal Expansion (for Constraints)	648
Roller	650
Symmetry	652
Symmetry Plane	654
Antisymmetry	656
Rigid Motion Suppression	657
Body Load	659
Gravity	660

Rotating Frame	662
Boundary Load	664
Edge Load	666
Point Load	667
Ring Load	668
Point Load (on Axis)	669
Spring Foundation	670
Thin Elastic Layer.	675
Predeformation	678
Added Mass.	680
Periodic Condition	682
Cell Periodicity	684
Boundary Pair	687
Low-Reflecting Boundary.	688
Thin-Film Damping	688
Rigid Connector	689
Center of Rotation Nodes	696
Thermal Expansion (Rigid Connector).	697
Applied Force (Rigid Connector).	698
Applied Moment (Rigid Connector).	700
Mass and Moment of Inertia (Rigid Connector)	701
Spring Foundation (Rigid Connector)	702
Attachment	704
Thermal Expansion (Attachment).	705
Beam Connection	707
Shell Connection	709
Contact	710
Friction	716
Adhesion	718
Bolt Pretension	721
Bolt Selection	722
Bolt Thread Contact	724
Thread Boundary Selection	725
Phase	726
Harmonic Perturbation	728
Stress Linearization	729
Important Variables in the Solid Mechanics Interface	731

Chapter 5: Shell and Plate

Theory for Shell and Plate Interfaces	736
About Shells and Plates	736
Theory Background for the Shell and Plate Interfaces	739
Theory for FSDT Laminated Shell	762
References for the Shell Interface	778
The Shell and Plate Interfaces	779
Domain, Boundary, Edge, Point, and Pair Nodes for the Shell and Plate Interfaces	783
Initial Values	785
Thickness and Offset	786
Linear Elastic Material	788
Layered Linear Elastic Material.	789
Shell Local System	792
Thermal Expansion (for materials)	793
Hygroscopic Swelling	795
Initial Stress and Strain.	796
External Stress.	798
Safety	800
Damping	801
Fixed Constraint	802
Prescribed Displacement/Rotation	803
Thermal Expansion (for constraints)	807
Prescribed Velocity	809
Prescribed Acceleration	811
Pinned	813
No Rotation	814
Simply Supported.	816
Symmetry	817
Symmetry Plane	820
Antisymmetry	822
Body Load	823
Face Load	824
Edge Load	826
Point Load	828

Ring Load	830
Spring Foundation	831
Predeformation	835
Added Mass.	835
Point Mass	837
Point Mass Damping	838
Periodic Condition	838
Solid Connection.	842
Beam Connection	843
Rigid Connector	846
Thermal Expansion (Rigid Connector).	848
Attachment	850
Thermal Expansion (Attachment).	850
Phase	852
Harmonic Perturbation	853

Chapter 6: Layered Shell

Theory for the Layered Shell Interface	856
About Layered Shells	856
Theory Background for the Layered Shell Interface	857
The Layered Shell Interface	869
The Layer Selection Section.	872
Domain, Boundary, Edge, Point, and Pair Nodes for the Layered Shell Interface.	872
Initial Values	875
Linear Elastic Material	876
Viscoelasticity	878
Creep	879
Viscoplasticity	880
Thermal Expansion (for Materials)	881
Hygroscopic Swelling	882
Initial Stress and Strain.	883
External Stress.	884
External Strain.	885

Safety	886
Damping	886
Rigid Domain	887
Free.	888
Prescribed Displacement	889
Prescribed Displacement, Interface	890
Prescribed Velocity	892
Prescribed Velocity, Interface	893
Prescribed Acceleration	894
Prescribed Acceleration, Interface	895
Fixed Constraint	896
Fixed Constraint, Interface	897
Thermal Expansion (for Constraints)	898
Roller	898
Roller, Interface	899
Symmetry	900
Antisymmetry	901
Rigid Motion Suppression	902
Body Load	903
Face Load	904
Gravity	905
Rotating Frame	905
Boundary Load	906
Edge Load	907
Line Load.	908
Point Load	908
Phase	909
Spring Foundation	910
Spring Foundation, Interface	911
Thin Elastic Layer.	912
Thin Elastic Layer, Interface	913
Predeformation	914
Added Mass.	914
Added Mass, Interface	915
Rigid Connector	916
Rigid Connector, Interface	917
Thermal Expansion (Rigid Connector).	918
Attachment	919

Thermal Expansion (Attachment).	920
Continuity	921

Chapter 7: Membrane

Theory for the Membrane Interface	924
About Membranes	924
Theory Background for the Membrane Interface	925
The Membrane Interface	929
Boundary, Edge, Point, and Pair Nodes for the Membrane Interface.	931
Linear Elastic Material	933
Initial Stress and Strain.	936
External Stress.	938

Chapter 8: Beam

Theory for the Beam Interface	942
About Beams	942
In-Plane Beams	943
3D Beam	944
Geometric Variables	945
Shape Functions	946
Geometric Nonlinearity	948
Strain-Displacement/Rotation Relation.	951
Stress-Strain Relation	951
Thermal Strain.	952
Hygroscopic Swelling	952
Initial Load and Strain	953
Implementation	954
Stress Evaluation	956
Common Cross Sections.	958
The Beam Interface	975
Boundary, Edge, Point, and Pair Nodes for the Beam Interface	978

Initial Values	980
Cross Section Data	980
Section Orientation	985
Linear Elastic Material	988
Thermal Expansion (for Materials)	990
Hygroscopic Swelling	991
Initial Stress and Strain.	992
External Stress.	993
Prescribed Displacement/Rotation	994
Prescribed Velocity	997
Prescribed Acceleration	998
Pinned	1000
Thermal Expansion (for constraints)	1001
No Rotation	1002
Symmetry	1003
Antisymmetry	1005
Edge Load	1006
Point Load	1008
Point Mass	1009
Point Mass Damping	1010
Solid Connection.	1011
Shell Connection	1012
Rigid Connector	1013
Attachment	1016
Phase	1016
Harmonic Perturbation	1017

Chapter 9: Beam Cross Section

Using the Beam Cross Section Interface	1020
About Beams and Cross Section Data	1020
Using the Beam Cross Section Interface	1021
Theory for the Beam Cross Section Interface	1026
Cross Section Properties.	1026
Computation of Stresses	1040

The Beam Cross Section Interface	1043
Hole	1044
Safety	1045

Chapter 10: Truss

Modeling with Truss Elements	1048
Theory for the Truss Interface	1050
Theory Background for the Truss Interface	1050
The Truss Interface	1058
Boundary, Edge, Point, and Pair Nodes for the Truss Interface	1059
Cross Section Data	1061
Straight Edge Constraint	1062
Linear Elastic Material	1063
Thermal Expansion (for materials)	1064
Hygroscopic Swelling	1065
Initial Stress and Strain.	1066
External Stress.	1067
Spring-Damper	1068
Pinned	1069
Thermal Expansion (for constraints)	1070
Symmetry	1072
Antisymmetry	1074
Edge Load	1075
Point Mass	1076
Point Mass Damping	1077
Phase	1077

Chapter 11: Multiphysics Interfaces and Couplings

The Thermal Stress Interface	1081
The Fluid-Solid Interaction Interface	1084
The Fluid-Shell Interaction Interface	1087
The Fluid-Membrane Interaction Interface	1090
The Fluid-Solid Interaction, Fixed Geometry Interface	1093
The Fluid-Shell Interaction, Fixed Geometry Interface	1095
The Fluid-Membrane Interaction, Fixed Geometry Interface	1098
The Joule Heating and Thermal Expansion Interface	1101
The Piezoelectric Devices Interface	1105
Coupling Piezoelectric Devices with Acoustics	1108
The Magnetostriction Interface	1109
Hygroscopic Swelling Coupling	1112
Couplings Between Structural Mechanics Interfaces	1113
Multiphysics Couplings	1114
Thermal Expansion (Multiphysics Coupling)	1114
Thermal Expansion, Layered Shell (Multiphysics Coupling)	1117
Hygroscopic Swelling (Multiphysics Coupling)	1120
Piezoelectric Effect	1121
Magnetostriction	1123
Fluid-Structure Interaction	1124
Fluid-Structure Interaction, Pair	1126
Fluid-Structure Interaction, Fixed Geometry	1128
Solid-Shell Connection.	1128

Solid-Beam Connection	1130
Shell-Beam Connection	1134

Chapter 12: Glossary

Glossary of Terms	1138
--------------------------	-------------

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, and trusses.

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:

- [Why Structural Mechanics is Important for Modeling](#)
- [What Problems Can It Solve?](#)
- [The Structural Mechanics Module Physics Interface Guide](#)
- [Common Physics Interface and Feature Settings and Nodes](#)
- [Geometry Levels for Study Capabilities](#)
- [Where Do I Access the Documentation and Application Libraries?](#)



[The Physics Interfaces and Building a COMSOL Multiphysics Model in the *COMSOL Multiphysics Reference Manual*](#)

Why Structural Mechanics is Important for Modeling

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 and plane strain, axial symmetry, and 3D solids
- Beams in 2D and 3D, Euler and Timoshenko theory
- Truss and cable elements
- Shells and plates, Mindlin theory
- Membranes

The module's study capabilities include static, eigenfrequency, time dependent (transient), frequency response, buckling, 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, 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.

The module is a collection of physics interfaces for COMSOL Multiphysics that handles static, eigenfrequency, transient, frequency response, parametric, transient thermal stress, and other analyses for applications in structural mechanics, solid mechanics, and piezoelectricity.

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.

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, 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 elastic 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 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 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 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.



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
 Fluid Flow				
Fluid-Solid Interaction ²		—	3D, 2D, 2D axisymmetric	stationary; stationary, one-way coupled; time dependent; time dependent, one-way coupled
Fluid-Solid Interaction, Fixed Geometry ²		—	3D, 2D, 2D axisymmetric	stationary; time dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Fluid-Shell Interaction ²		—	3D, 2D axisymmetric	stationary; stationary, one-way coupled; time dependent; time dependent, one-way coupled
Fluid-Shell Interaction, Fixed Geometry ²		—	3D, 2D axisymmetric	stationary; time dependent
Fluid-Membrane Interaction ²		—	3D, 2D axisymmetric	stationary; stationary, one-way coupled; time dependent; time dependent, one-way coupled
Fluid-Membrane Interaction, Fixed Geometry ²		—	3D, 2D axisymmetric	stationary; time dependent
 Structural Mechanics				
Solid Mechanics ¹		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; eigenfrequency, prestressed; mode analysis; time dependent; time dependent, modal; time dependent, prestressed, modal; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; modal reduced order model; response spectrum, linear buckling; bolt pre-tension
Thermal Stress ²		—	3D, 2D, 2D axisymmetric	stationary; time dependent
Joule Heating and Thermal Expansion ²		—	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; frequency domain; frequency domain, modal; frequency domain, prestressed; frequency domain, prestressed, modal; modal reduced order model; response spectrum; linear buckling
Plate		plate	2D	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; modal reduced order model; response spectrum; linear buckling
Beam		beam	3D, 2D	stationary; eigenfrequency; frequency domain; frequency domain, modal; time dependent; time dependent, modal; modal reduced order model; linear buckling
Beam Cross Section		bcs	2D	stationary

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Truss		truss	3D, 2D	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; modal reduced order model; response spectrum; linear buckling
Membrane		mbrn	3D, 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; modal reduced order model; response spectrum
Piezoelectric Devices ²		—	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; modal reduced order model; small-signal analysis; ; linear buckling

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
Magnetostriction ²		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; eigenfrequency, prestressed; frequency domain, prestressed
¹ This physics interface is included with the core COMSOL package but has added functionality for this module.				
² This physics interface is a predefined multiphysics coupling that automatically adds all the physics interfaces and coupling features required.				

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 2-3](#) for links to common sections and [Table 2-4](#) 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.

Geometry Levels for Study Capabilities

The column for the dependent variables shows the field variables that formulate the underlying equations. Depending on the engineering assumptions and the geometry dimension, these variables include a subset of the displacement field u , v , and w in the global coordinate system, pressure, and temperature. The Shell and Plate interfaces use as dependent variables the variables a_x , a_y , and a_z , which are the displacements of the shell normals in the global x , y , and z directions, respectively. Such variables can be expressed in terms of customary rotations ϕ_x , ϕ_y , and ϕ_z about the global axes.

For each physics interface, the table indicates dependent variables and the geometry levels (where data such as material properties, loads, and constraints are specified). Edges exist only in 3D geometries. Studies are listed in a separate table in [The Structural Mechanics Module Physics Interface Guide](#) section.



Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

PHYSICS INTERFACE	NAME	DEPENDENT VARIABLES	GEOMETRY LEVEL			
			POINTS	EDGES	BOUNDARIES	DOMAINS
STRUCTURAL MECHANICS						
Solid Mechanics	solid	$\mathbf{u}, (p)$	√	√	√	√
Shell	shell	\mathbf{u}, \mathbf{a}	√	√	√	
Plate (3 DOF)	plate	w, a_x, a_y	√		√	√
Plate (6 DOF)	plate	\mathbf{u}, \mathbf{a}	√		√	√
Beam	beam	\mathbf{u}, θ	√	√		
Truss	truss	\mathbf{u}	√	√		
Membrane	mbrn	\mathbf{u}	√	√	√	
Thermal Stress	n/a*	$\mathbf{u}, (p), T$	√	√	√	√
Joule Heating and Thermal Expansion	n/a*	$\mathbf{u}, (p), T, V$	√	√	√	√
Piezoelectric Devices	n/a*	\mathbf{u}, V	√	√	√	√
Fluid Flow						
Fluid-Structure Interaction	fsi	$\mathbf{u}_{\text{solid}}, \mathbf{u}_{\text{fluid}}, p$	√	√	√	√
Fluid-Structure Interaction, Fixed Geometry	n/a*	$\mathbf{u}_{\text{solid}}, \mathbf{u}_{\text{fluid}}, p$	√	√	√	√

* These are multiphysics couplings that do not have names.

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*.



Win

To open the **Help** window:

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

Mac	<p>To open the Help window:</p> <ul style="list-style-type: none"> • In the Model Builder or Physics Builder click a node or window and then press F1.
Linux	<ul style="list-style-type: none"> • On the main toolbar, click the Help () button. • From the main menu, select Help>Help.

Opening the Documentation Window

Win	<p>To open the Documentation window:</p> <ul style="list-style-type: none"> • Press Ctrl+F1. • From the File menu select Help>Documentation ().
-----	--

Mac	<p>To open the Documentation window:</p> <ul style="list-style-type: none"> • Press Ctrl+F1.
Linux	<ul style="list-style-type: none"> • On the main toolbar, click the Documentation () button. • From the main menu, select Help>Documentation.

THE APPLICATION LIBRARIES WINDOW

Each model or application includes documentation with the theoretical background and step-by-step instructions to create a model or app. 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.

	<p>The Application Libraries Window in the <i>COMSOL Multiphysics Reference Manual</i>.</p>
---	---

Opening the Application Libraries Window

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

	<ul style="list-style-type: none">• From the Home toolbar, Windows menu, click () Applications Libraries.• From the File menu select Application Libraries. <p>To include the latest versions of model examples, from the File>Help menu, select () Update COMSOL Application Library.</p>
	<p>Select Application Libraries from the main File> or Windows> menus.</p>
	<p>To include the latest versions of model examples, from the Help menu select () Update COMSOL Application Library.</p>

CONTACTING COMSOL BY EMAIL

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

COMSOL ACCESS AND TECHNICAL SUPPORT

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and a case number 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	www.comsol.com
Contact COMSOL	www.comsol.com/contact
COMSOL Access	www.comsol.com/access
Support Center	www.comsol.com/support
Product Download	www.comsol.com/product-download
Product Updates	www.comsol.com/support/updates
COMSOL Blog	www.comsol.com/blogs
Discussion Forum	www.comsol.com/community
Events	www.comsol.com/events
COMSOL Video Gallery	www.comsol.com/video
Support Knowledge Base	www.comsol.com/support/knowledgebase

Overview of the User's Guide

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



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

TABLE OF CONTENTS, GLOSSARY, AND INDEX

To help you navigate through this guide, see the [Contents](#), [Glossary](#), and [Index](#).

MODELING WITH THE STRUCTURAL MECHANICS MODULE

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

STRUCTURAL MECHANICS THEORY

The [Structural Mechanics Theory](#) chapter introduces the general theory on which the physics interfaces in the Structural Mechanics Module are based.

THE SOLID MECHANICS INTERFACE

The [Solid Mechanics](#) chapter describes [The Solid Mechanics Interface](#), which is used to model 3D solids, plane strain and plane stress 2D 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 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 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 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 Piezoelectric Devices Interface](#) combines a Solid Mechanics with an Electrostatics interface. Piezoelectric materials in 3D, 2D plane strain and plane stress, and axial symmetry can be modeled.

- [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](#), found under the Fluid Flow branch, 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](#), found under the Fluid Flow branch, 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](#), found under the Fluid Flow branch, 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](#), found under the Fluid Flow branch, 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 non-deforming. The solid material exists on domains which are adjacent to the fluid.
- [The Fluid-Shell Interaction, Fixed Geometry Interface](#), found under the Fluid Flow branch, 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 non-deforming. The shell is modeled on the boundary of the fluid.
- [The Fluid-Membrane Interaction, Fixed Geometry Interface](#), found under the Fluid Flow branch, 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 non-deforming. The membrane is modeled on the boundary of 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 <http://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
- Mechanical Damping and Losses
- Modeling Geometric Nonlinearity
- Contact Modeling
- Activating and Deactivating Material
- Springs and Dampers
- Defining Multiphysics Models
- Thermally Coupled Problems
- Computing Mass Properties
- Pretensioned Bolts
- Performing a Response Spectrum Analysis
- Stress Linearization
- Solver Settings
- 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](#)
- [Modal Superposition](#)
- [Harmonic Perturbation](#)
- [Modal Reduced Order](#)
- [Linearized Buckling Analysis](#)
- [Bolt Pretension 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 one-way coupled in the sense 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, λ , or the eigenfrequency, f , which is more commonly used in a structural mechanics context. The relation between the two is

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

where i is the imaginary unit.

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. **Scaling of eigenvectors** is set to **Mass matrix**, the eigenmodes \mathbf{U} are orthogonalized with respect to the mass matrix \mathbf{M} so that

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

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 modal 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 modal 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 which would be obtained if mass matrix scaled eigenmodes would have been used.



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.



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

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**.



In the *COMSOL Multiphysics Reference Manual*:

- [Eigenvalue Solver](#)
- [Studies and Solvers](#)
- [Derived Values, Evaluation Groups, and Tables](#)

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

- [Modal Participation Factors](#)

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.

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 ratio between the imaginary and real parts of the eigenfrequency is the relative damping of the corresponding eigenmode,

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

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

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 modal 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 the 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 not all loads have the same phase, you can prescribe the phase of a certain 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 amplitude and phase information is not reliable.



Some extra variables for postprocessing are created in a frequency-domain analysis. As an example, in the Solid Mechanics interface 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 X direction
- `solid.uAmp_tX`—Amplitude of velocity in X direction
- `solid.uAmp_ttX`—Amplitude of acceleration in 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 provide as 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 an 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.**

Modal Superposition

Analyzing forced dynamic response for large models can be very time-consuming. You can often improve the performance dramatically by using the modal 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 modal 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 modal 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 modal 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 modal superposition formulation in COMSOL, 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 modal 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 modal 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 modal 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 modal 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 into account for example stiffness from follower loads, 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 an **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

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, 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

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

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{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 2-2. First substitute for \mathbf{u} using Equation 2-6. Then transform the equation to the *modal coordinate system* by premultiplying by U^T . This gives:

$$U^T M U \ddot{\mathbf{a}} + U^T D U \dot{\mathbf{a}} + U^T K U \mathbf{a} = U^T \mathbf{F} \quad (2-7)$$

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 α and β are constants. For a general damping, the transformed damping matrix is however not diagonal. As an alternative, a damping ratio, ζ_i , can be assigned to each mode.

EIGENVALUE SCALING

The precise form of Equation 2-7 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 2-3 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 ω_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 2-7 takes the form

$$\ddot{\mathbf{a}} + \text{diag}(2\zeta_i \omega_i) \dot{\mathbf{a}} + \text{diag}(\omega_i^2) \mathbf{a} = U^T \mathbf{F} \quad (2-8)$$

It is also possible to scale the eigenvectors so that the point of maximum displacement has unit 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 2-7 takes the form

$$\text{diag}(m_{\text{eff},i})\ddot{\mathbf{a}} + \text{diag}(c_{\text{eff},i})\dot{\mathbf{a}} + \text{diag}(k_{\text{eff},i})\mathbf{a} = U^T\mathbf{F} \quad (2-9)$$

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 $\sqrt{3}$.

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 1.

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

$$U'^T M U' \ddot{\mathbf{a}}' + U'^T D U' \dot{\mathbf{a}}' + U'^T K U' \mathbf{a}' = U'^T \mathbf{F} \quad (2-10)$$

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 make the assumption 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:

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

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.

CONSTRUCTING REDUCED-ORDER MODELS WITH COMSOL

To obtain the data necessary to construct reduced-order models from a COMSOL simulation the **Modal Reduced Order Model** study step is available. Add this study step after an existing Eigenvalue study step, by right-clicking the **Study I** node and selecting **Study Steps>Time Dependent>Modal Reduced Order Model**. Then solve the model. 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. The Modal Reduced Order Model exports these matrices in a format that respects the normalization of the preceding Eigenvalue study. To change this select the **Study I>Solver Configurations>Solver I>Eigenvalue Solver I** 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. So if the linear problem is solved first for an arbitrary initial load level \mathbf{f}_0 ,

$$\mathbf{K}_L\mathbf{u}_0 = \mathbf{f}_0$$

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 λ 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 λ .

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

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

The 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 λ is smaller than 1.

	<p>Do not select geometric nonlinearity in a linearized buckling analysis. Since the nonlinear effect of the stress is already taken into account in the formulation, an explicit use of geometric nonlinearity would make the computed buckling load dependent on the load level used in the preload study step.</p>
	<p>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 <i>imperfection sensitivity</i>. 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.</p>
	<p>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.</p>
	<ul style="list-style-type: none">• Studies and Solvers and Linear Buckling in the <i>COMSOL Multiphysics Reference Manual</i>• Linear Buckling in the theory section of the <i>Structural Mechanics Module User's Guide</i>
	<ul style="list-style-type: none">• <i>Bracket — Linear Buckling Analysis</i>: Application Library path Structural_Mechanics_Module/Tutorials/bracket_linear_buckling• <i>Linear Buckling Analysis of a Truss Tower</i>: Application Library path Structural_Mechanics_Module/Buckling/truss_tower_buckling

Bolt Pretension Study

The **Bolt Pretension** study step is a special case of a **Stationary** study step, where the special degrees of freedoms used for modeling prestressed bolts are solved for. In all

other study types, these degrees of freedom are inactive. Typically, you include a **Bolt Pretension** study step as the first step in a study in order to simulate the state after the assembly of a bolted joint. You can add any other types of study steps for computing the effects of the service loads.



Pretensioned Bolts



Prestressed Bolts in a Tube Connection: Application Library path
Structural_Mechanics_Module/Contact_and_Friction/tube_connection

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** data sets.

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 data sets are described in [Response Spectrum 2D and Response Spectrum 3D](#) in the *COMSOL Multiphysics Reference Guide*.
 - The [Response Spectrum](#) node is described in the *COMSOL Multiphysics Reference Guide*.
-



- *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
- **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
- **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 <http://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.

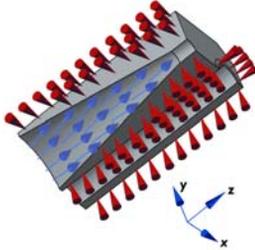


Figure 2-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.

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.

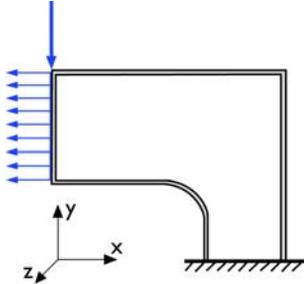


Figure 2-2: Plane stress models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

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 ϵ_z , ϵ_{yz} , and ϵ_{xz} are zero.

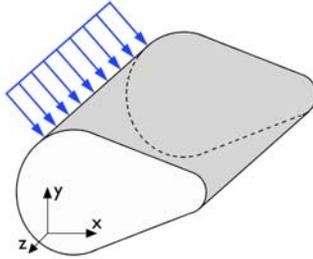


Figure 2-3: A geometry suitable for plane strain analysis.

Loads in the x and y directions are allowed. The loads are assumed to be constant throughout the thickness of the material, but the thickness can vary with x and y . 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 , ϕ (ϕ i), and z . Loads are independent of ϕ , 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.

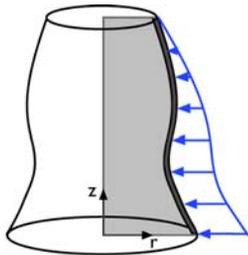


Figure 2-4: Rotating a 2D geometry to recover a 3D solid.

Shell and Plate

The Shell interface is useful when the object is thin in one direction. Structures built from welded or bolted flat plates is an archetypal shell structure, 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 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 shear deformations are taken into account. This allows the Beam interface to be applied to rather thick beams.

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 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](#)
- [Solver Settings](#)

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

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

Coupling Techniques

The following basic techniques to connect physics interfaces with displacement degrees of freedom is 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 are 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 \(3D\)](#)
- [Beam Point to Solid Boundary \(2D\)](#)

- [Beam Point to Solid Boundary \(3D\)](#)
- [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 \(3D\)](#)

Shell Edge to Solid Boundary (3D)

A shell can be coupled to a solid by adding a [Solid-Shell 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 which 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-Shell Connection](#) multiphysics coupling with **Connection type** set to **Solid and shell shared boundaries** or **Solid and shell 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.

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 coupling 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 Boundary (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 around the beam end that is connected in several 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 includes 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.

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 **Solid and beam shared boundaries** or **Solid and beam 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 which 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 **Shell and beam shared boundaries** or **Shell and beam 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 around the beam end that is connected in several 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 how large portion of the edge that is connected to the beam end in several ways.

The underlying theory and more details can be found in [Connection Between Shells and Solids](#) and [Connection Between Shells and Beams](#).



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

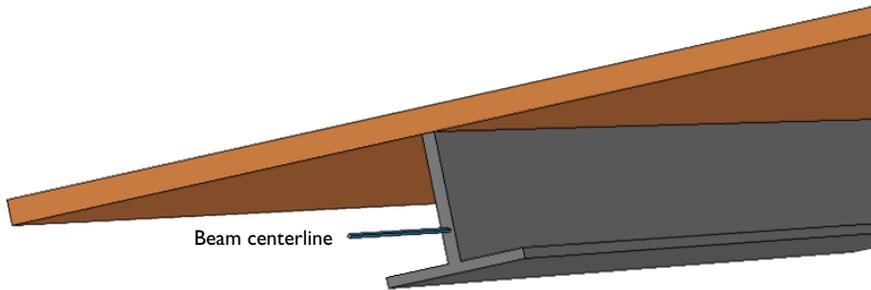
USING GENERAL COUPLING OPERATORS

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

An example could be a shell stiffened by beams. In practice, you would probably use the built-in coupling described in [Beam Edge to Shell Edge \(3D\)](#) for this case, but the examples 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 ϕ is the rotation vector, which contains the rotational degrees of freedom in the Beam interface. The rotation vector is also available as a variable in the Shell interface, where it is derived from the rotational degrees of freedom \mathbf{a} . The shell normal is denoted by \mathbf{n} .

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 `genext1(u)+genext1(shell.thy)*zdist`, where `zdist` is some expression defining the distance from the beam axis to the shell midsurface.



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



In the *COMSOL Multiphysics Reference Manual*:

- [Component Couplings and Coupling Operators and General Extrusion](#)
 - [About Component Couplings](#)
-

Applying Loads

An important aspect of structural analysis is the formulation of the forces applied to the modeled structure. The freedom is available to 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. Add [The Solid Mechanics Interface](#) () to the Model Builder, then from the **Physics** toolbar, **Domains**, **Boundaries**, **Edges**, and **Points** menus, click to select **Body Load**, **Face Load**, **Edge Load**, or **Point Load**. This guide includes a detailed description of the functionality for each physics interface.

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, $100[\text{lbf}/\text{in}^2]$.

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.

You can always display the loads actually used after the analysis, since they are available among the result quantities. 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 can then do like this:

- 1 On the **Study** toolbar, click **Get Initial Value**. This operation is fast when compared to actually solving the problem.
- 2 From the **Results** toolbar, add a **3D Plot Group** or **2D Plot Group**.
- 3 Add a suitable plot type to the plot group, for example an **Arrow Surface** plot.

- 4 Select a result from the **Load** group in the results for the physics interface.

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none">• Physics Symbols• Using Units• Coordinate Systems
	<p>For an example showing how to examine the load distribution, see <i>Bracket — Static Analysis</i>: Application Library path Structural_Mechanics_Module/Tutorials/bracket_static. This is also the first example used in the <i>Introduction to the Structural Mechanics Module</i>.</p>

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.
- 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.



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/Civil_Engineering/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 are in agreement 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.

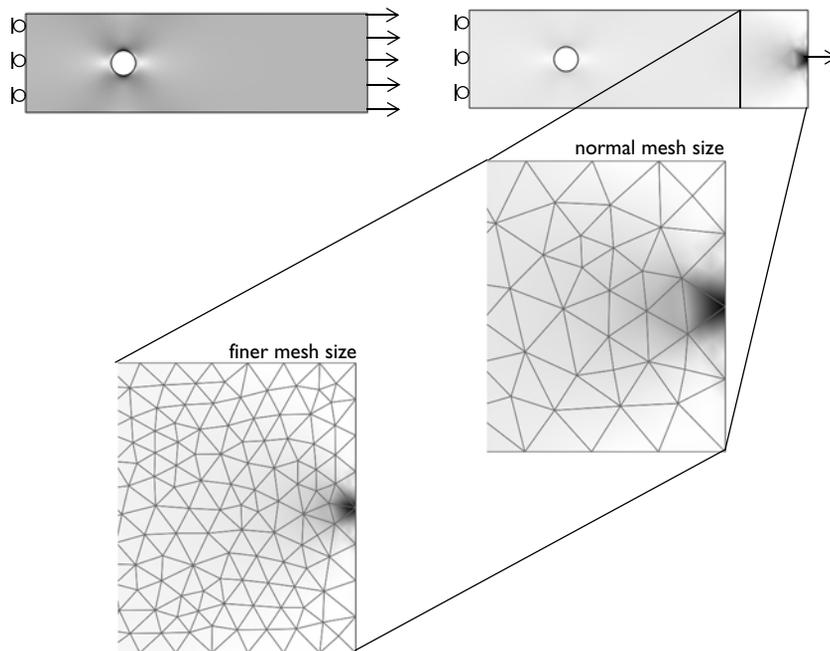


Figure 2-5: A plate with a hole subject to a distributed load (left) and a point load (right).

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, 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

Acceleration loads can be found, for example, in the structural mechanics analysis of an airplane seat. Acceleration or deceleration of the aircraft produces a force that an accurate simulation must include. Because expressions can be used when specifying loads, it is easy to model acceleration loads.

For modeling rotating parts under static conditions, use centrifugal acceleration loads. The body load in the radial direction is

$$K_r = \rho\omega^2r \quad (2-11)$$

where ρ is the density of the material, ω is the angular frequency, and r is the radial distance from the axis of rotation. A cylindrical coordinate system is often useful here.

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 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 its acts.

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 problems 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 2-4: 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 is 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.

In frequency domain, a prescribed velocity v_p or prescribed acceleration a_p can be directly interpreted as a prescribed displacement u_p :

$$u_p = \frac{v_p}{i\omega}$$

$$u_p = \frac{-a_p}{\omega^2}$$

where ω is the angular frequency.

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 a known a priori and does not depend on the solution itself, this is possible.

- When the velocity or acceleration has a simple time dependence, you can integrate it analytically one or two times to obtain the displacement, and directly prescribe the displacement instead.
- When you have complicated known velocity or acceleration histories, for example from measurements, you can use the `integrate()` operator. In this case, you enter the prescribed displacement as `integrate(my_data(tau), tau, 0, t)`. Here `my_data` is the measured data as function of time, and `tau` is a dummy integration variable

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.

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.

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.

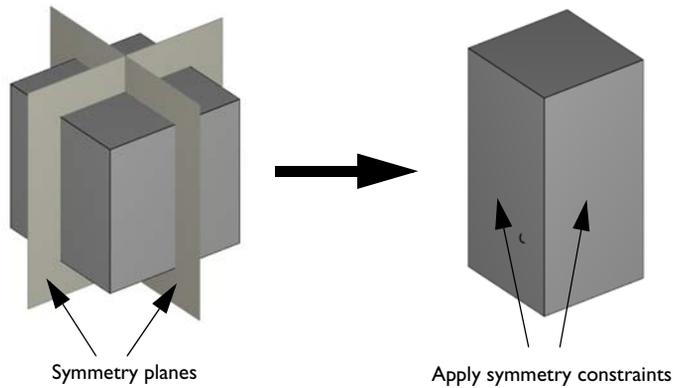


Figure 2-6: If the geometry exhibits two symmetry planes, model a quarter of the geometry by using the *Symmetry* feature for the two selected surfaces.



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.

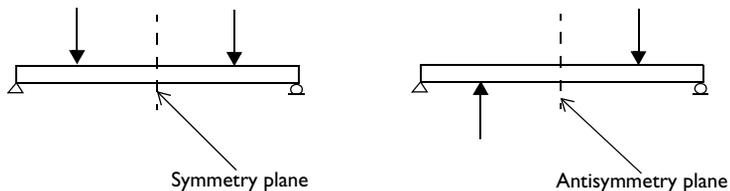


Figure 2-7: *Symmetry plane (left) and antisymmetry plane (right).*

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](#).

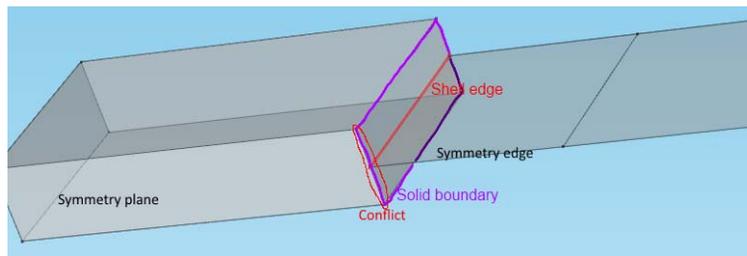


Figure 2-8: Example of potentially conflicting constraints

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>Component 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` or `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` or `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 Internal 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 internal boundaries. On internal 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 up and down side of the boundary respectively. If you need the value of the total force acting on an internal section through your model, these variable 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 internal 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 2-5 you can find an overview of the families of materials, and their applicability in the various structural mechanics interfaces.

TABLE 2-5: MATERIAL MODELS IN THE BASIC PHYSICS INTERFACES

MATERIAL MODEL	SOLID MECHANICS	SHELL/ PLATE	MEMBRANE	BEAM	TRUSS
Linear Elastic Material	X	X	X	X	X
Nonlinear Elastic Material	X	—	X	—	—
Hyperelastic Material	X	—	X	—	—
Shape Memory Alloy	X	—	—	—	—
Piezoelectric Material	X	—	—	—	—
Magnetostrictive Material	X	—	—	—	—
Viscoelasticity	X	—	X	—	—
Plasticity	X	—	X	—	X
Soil Plasticity	X	—	—	—	—
Creep	X	—	X	—	—
Elastoplastic Soil Material	X	—	—	—	—
Viscoplasticity	X	—	X	—	—
Porous Plasticity	X	—	—	—	—
Concrete	X	—	—	—	—
Rocks	X	—	—	—	—
Damage	X	—	—	—	—
External Stress-Strain Relation	X	—	—	—	—
Rigid Domain	X	—	—	—	—
Spring-Damper	—	—	—	—	X

Many of the material models can be augmented by effects like thermal expansion, hygroscopic swelling, initial stresses and strains, and external stress.

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 ν_{xy} by

$$\nu_{xy} = \nu_{yx} \frac{E_x}{E_y}$$



In the theory section [Linear Elastic Material](#)

Introduction to Viscoelastic Materials

The generalized Maxwell, standard linear solid (SLS) and Kelvin-Voigt viscoelastic materials are available. All the models are linear, and the corresponding materials can be described as consisting of one or more branches with a spring and a dashpot acting in parallel to a linear elastic material. For each viscoelastic branch, the shear modulus and the relaxation time (or viscosity) are entered.



- [Linear 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.

When the **Pressure formulation** is selected in the **Use mixed formulation** list, the negative volumetric stress p_w is treated as an additional dependent variable. The resulting mixed formulation is also known as u-p formulation. This formulation removes the effect of 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 a finite-element sense) to the original pressure calculated from the strains.

Select this setting when the material data is close to incompressibility. For an isotropic 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.

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 similar result.

When using the crack band method or no regularization at all, the following steps are recommended:

- The size of the biggest mesh element should not exceed $h < 2EG_f/\sigma_{ts}^2$, where E is the Young's modulus, G_f is the fracture energy per unit area, and σ_{ts} is the tensile strength. Larger values of h will cause a snap-back of the stress-strain curve at the material point level.
- Use linear shape functions for the displacement. 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 round off.
- 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.

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 Materials Database](#)
- [Piezoelectric Materials Database](#)
- [Liquids and Gases Materials Database](#)
- [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 effective 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. Non-zero 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_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 effective stress

```

/*
* Function: csext_mises
* -----
* Description:
*   Computes and returns the von Mises effective 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 effective stress

```

/*
* Function: csext_mises_voigt
* -----
* Description:
*   Computes and returns the von Mises effective 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_utils: 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 Approaches](#) 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:

$$\begin{aligned} \mathbf{S} &= s_E \mathbf{T} + d^T \mathbf{E} \\ \mathbf{D} &= d \mathbf{T} + \epsilon_T \mathbf{E} \end{aligned} \quad (2-12)$$

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 ϵ_T , correspond to the material compliance, the coupling properties and the permittivity. These quantities are tensors of rank 4, 3, and 2

respectively. The tensors however are highly symmetric for physical reasons, and they can be represented as matrices within an abbreviated subscript notation, which is usually more convenient. In the Piezoelectric Devices interface, the Voigt notation is used, which is standard in the literature for piezoelectricity but which differs from the defaults in the Solid Mechanics interface.



The Piezoelectric Material uses the structural mechanics nomenclature. The strain is named ε (instead of \mathbf{S}) and the stresses are denoted by either σ or S (instead of \mathbf{T}). This makes the names consistent with those used in the other structural mechanics interfaces.

Equation 2-12 will, using the notation from structural mechanics, then read

$$\begin{aligned}\varepsilon &= s_E \sigma + d^T \mathbf{E} \\ \mathbf{D} &= d \sigma + \varepsilon_0 \varepsilon_{rT} \mathbf{E}\end{aligned}\tag{2-13}$$

Equation 2-12 (or Equation 2-13) 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:

$$\begin{aligned}\sigma &= c_E \varepsilon - e^T \mathbf{E} \\ \mathbf{D} &= e \varepsilon + \varepsilon_0 \varepsilon_{rS} \mathbf{E}\end{aligned}\tag{2-14}$$

The material properties, c_E , e , and ε_S are related to s_E , d , and ε_T . It is possible to use either form of the constitutive relations. In addition to Equation 2-12 or Equation 2-14, 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 2-6). 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: $(YXI) -35.25^\circ$. 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° in this convention and positive angles therefore correspond to the opposite direction of rotation (see [Figure 2-9](#)). [Table 2-7](#) 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 2-6](#) summarizes the signs for the important matrix elements under the two conventions. [Table 2-7](#) shows the different definitions of the crystal cuts under the two conventions.

TABLE 2-6: 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 2-7: 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	(<i>YXl</i>) +35.25°	(<i>YXl</i>) -49°
	Y-thickness Euler angles	(<i>ZXZ</i> : 0°, -35.25°, 0°)	(<i>ZXZ</i> : 0°, 49°, 0°)
	Z-thickness Euler angles	(<i>ZXZ</i> : 0°, -125.25°, 0°)	(<i>ZXZ</i> : 0°, -41°, 0°)
IEEE 1978	Standard	(<i>YXl</i>) -35.25°	(<i>YXl</i>) +49°
	Y-thickness Euler angles	(<i>ZXZ</i> : 0°, 35.25°, 0°)	(<i>ZXZ</i> : 0°, -49°, 0°)
	Z-thickness Euler angles	(<i>ZXZ</i> : 0°, -54.75°, 0°)	(<i>ZXZ</i> : 0°, -139°, 0°)

When defining the material orientation, it is necessary to consider the orientation of the plate with respect to the global coordinate system in addition to the orientation of the plate with respect to the crystallographic axes. Consider the example of AT cut quartz in [Figure 2-10](#). 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.

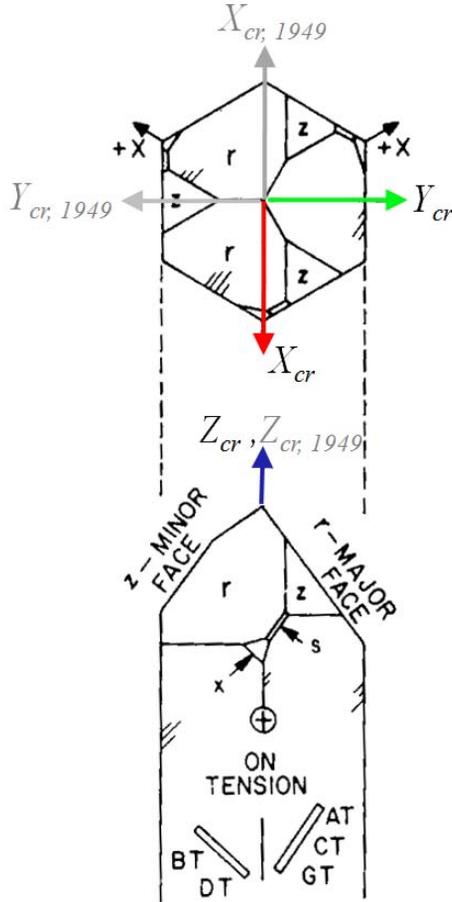


Figure 2-9: Crystallographic axes defined for right-handed quartz in COMSOL and the 1978 IEEE standard (color). The 1949 standard axes are shown for comparison (gray).



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.

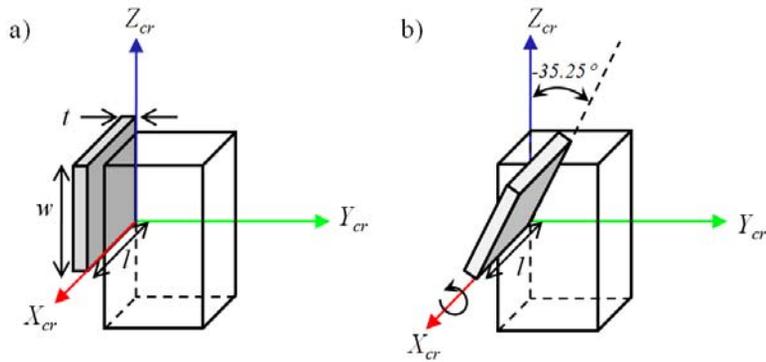


Figure 2-10: Definition of the AT cut of quartz within the IEEE 1978 standard. The AT cut is defined as: $(YXL) -35.25^\circ$. The first two bracketed letters specify the initial orientation of the plate, with the thickness direction, t , along the crystal Y axis and the length direction, l , along the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

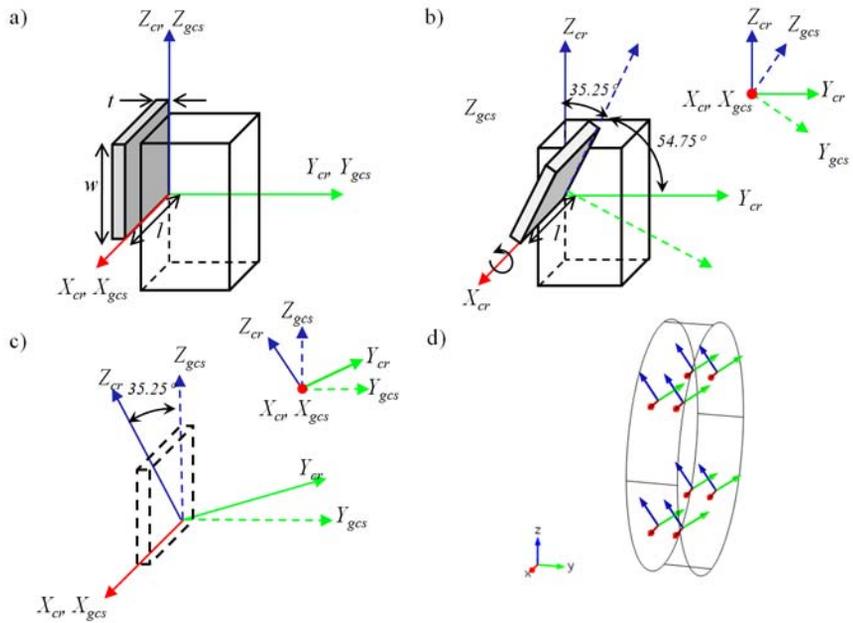


Figure 2-11: Defining an AT cut crystal plate within COMSOL, with normal in the global Y -direction. Within the 1978 IEEE standard the AT cut is defined as (YXL) -35.25° . Start with the plate normal or thickness in the Y_{cr} direction (a) and rotate the plate -35.25° about the l axis (b). The global coordinate system rotates with the plate. Finally rotate the entire system so that the global coordinate system is 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.

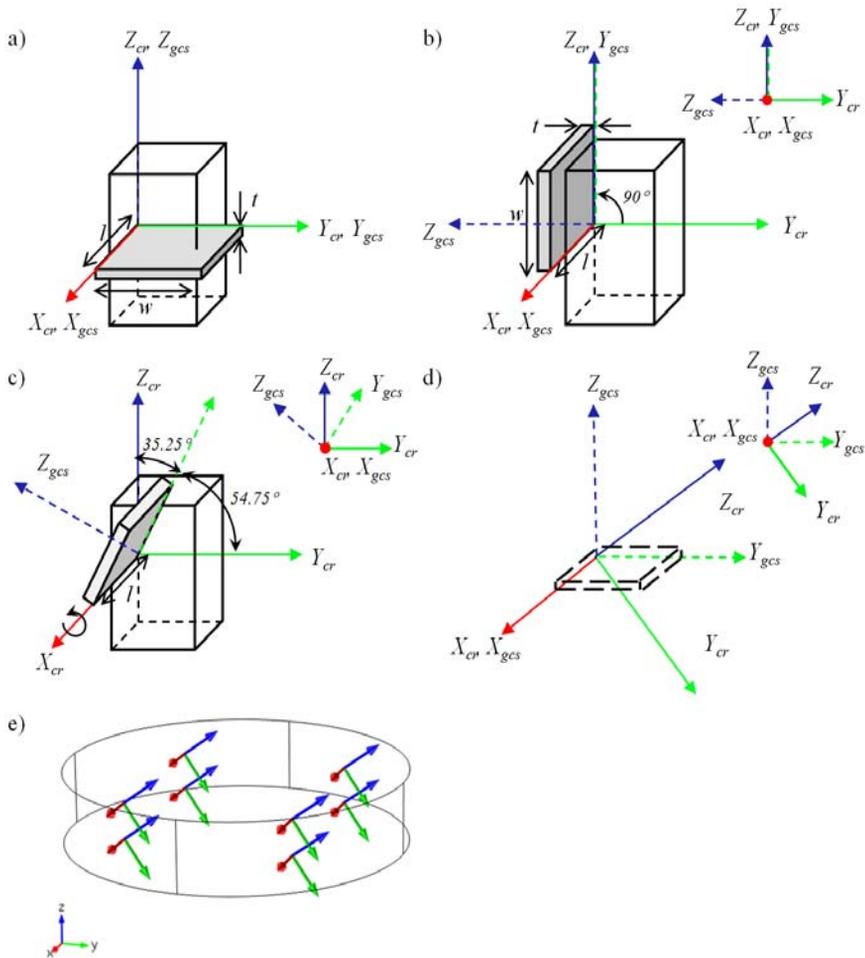


Figure 2-12: Defining an AT cut crystal plate within COMSOL, with normal in the global Z -direction. Within the 1978 IEEE standard the AT cut is defined as (YXl) -35.25° . Begin with the plate normal in the Z_{cr} -direction, so the crystal and global systems are coincident. Rotate the plate so that its thickness points in the Y_{cr} -direction (the starting point for the IEEE definition), the global system rotates with the plate (b). Rotate the plate -35.25° about the l axis (d). Finally rotate the entire system so that the global coordinate system is orientated as it appears in COMSOL (d). The local coordinate system should be defined with the Euler angles (ZXZ: 0, -54.75° , 0). (e) shows a coordinate system for this system in COMSOL.

Piezoelectric Losses

Losses in piezoelectric materials can be generated both mechanically and electrically.

In the frequency domain, these can be represented by introducing complex material properties in the elasticity and permittivity matrices, respectively. 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

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

where \tilde{c}_E , \tilde{d} , and $\boldsymbol{\varepsilon}$ 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, η_{cE} . For a structural material without coupling, simply use η_s , the structural loss factor.

The loss factors are defined so that a positive loss factor value usually corresponds to a positive loss. The complex-valued data is then based on sign rules.

By default, there is no damping until at least one of the damping and losses related subfeatures is added.

For the Piezoelectric Material node, the following equations apply via the corresponding three subnodes:

Mechanical Damping

$$\begin{aligned}\tilde{c}_E^{m,n} &= c_E^{m,n} (1 + j\eta_{cE}^{m,n}) \\ \tilde{s}_E^{m,n} &= s_E^{m,n} (1 - j\eta_{sE}^{m,n})\end{aligned}$$

where m and n refer to components of each matrix.

Coupling Loss

$$\begin{aligned}\tilde{e}^{m,n} &= e^{m,n} (1 + j\eta_e^{m,n}) \\ \tilde{d}^{m,n} &= d^{m,n} (1 + j\eta_d^{m,n})\end{aligned}$$

Dielectric Loss

$$\begin{aligned}\tilde{\epsilon}_{rS}^{m,n} &= \epsilon_{rS}^{m,n} (1 - j\eta_{\epsilon S}^{m,n}) \\ \tilde{\epsilon}_{rT}^{m,n} &= \epsilon_{rT}^{m,n} (1 - j\eta_{\epsilon T}^{m,n})\end{aligned}$$

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 σ_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 τ_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:

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

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

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

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 ϵ_{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 Piezoelectric Devices 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 **Piezoelectric Devices** from the **Model Wizard**,
- By choosing **Piezoelectric Devices** 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. Non-piezoelectric domains can be modeled using the 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 an other 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 cE , a loss factor for sE (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 ϵS , a loss factor for ϵT (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 ρ (SI unit: kg/m^3)
- Elasticity matrix cE (SI unit: Pa) in Stress-charge form.
- Compliance Matrix sE (SI unit: $1/\text{Pa}$) in Strain-charge form.

ELECTROSTATIC PROPERTIES:

- Relative Permittivity ϵ_{S} (dimensionless) in Stress-charge form.
- Relative Permittivity ϵ_{T} (dimensionless) in Strain-charge form.

COUPLING PROPERTIES

- Coupling matrix eES (SI unit: C/m^2) in Stress-Charge form.
- Coupling matrix dET (SI unit: C/N) in Strain-charge form.

DAMPING AND LOSS PROPERTIES

- Loss factor for elasticity matrix cE , η_{cE} , or loss factor for elasticity matrix sE , η_{sE} : required when Mechanical Damping is present. The latter is valid only in Strain-Charge form.
- Loss factor for electrical permittivity matrix ϵS , $\eta_{\epsilon S}$, or Loss factor for electrical permittivity matrix ϵT , $\eta_{\epsilon T}$: required when Dielectric Loss is present. The latter is valid only in Strain-Charge form.
- Loss factor for coupling matrix e , η_{eS} , or loss factor for coupling matrix d , η_{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 Database](#) in the *COMSOL Multiphysics Reference Manual*.

Coupling Piezoelectric Devices 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 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 ϵ , magnetic field \mathbf{H} , and magnetic flux density \mathbf{B} in either a *stress-magnetization form* or *strain-magnetization form*. In COMSOL Multiphysics, both constitutive forms can be used; simply select one, and the software will make all necessary transformations if needed. For details, see the corresponding the 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 the 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 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:

$$m \frac{d^2 u}{dt^2} + c \frac{du}{dt} + ku = f(t) \quad (2-15)$$

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:

$$\frac{d^2 u}{dt^2} + 2\zeta\omega_0 \frac{du}{dt} + \omega_0^2 u = \frac{f(t)}{m} \quad (2-16)$$

where $\xi = c/(2m\omega_0)$ and $\omega_0^2 = k/m$. In this case ζ is the *damping ratio* ($\zeta = 1$ for critical damping) and ω_0 is the *undamped resonant frequency* of the system. In the literature it is more common to give values of ζ than c . The relative damping ζ can also be readily related to many of the various measures of damping employed in different disciplines. These are summarized in [Table 2-8](#).

TABLE 2-8: 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 t_0 is a reference time and τ is the period of vibration for a decaying, unforced degree of freedom.</p>	$\delta_d \approx 2\pi\zeta$ <p>($\zeta \ll 1$)</p>

TABLE 2-8: 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 ω is the angular frequency and the amplitude terms U and F can in general be complex (the arguments provide information on the relative phase of signals). Usually the real part is taken as implicit and is subsequently dropped.

Equation 2-15 takes the following form in the frequency domain:

$$-\omega^2 m U + j\omega c U + k U = F \tag{2-17}$$

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

$$-\omega^2 U + 2j\omega\omega_0\xi U + \omega_0^2 U = \frac{F}{m} \tag{2-18}$$

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 2-15](#). This means that the method can be applied generally in either the time or frequency domain. The parameter c in [Equation 2-15](#) is defined as a fraction of the mass and the stiffness using two parameters, α_{dM} and β_{dK} , such that

$$c = \alpha_{dM}m + \beta_{dK}k \quad (2-19)$$

Substituting this relationship into [Equation 2-15](#) and rearranging into the form of [Equation 2-16](#) 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:

$$\zeta = \frac{1}{2}\left(\frac{\alpha_{dM}}{\omega_0} + \beta_{dK}\omega_0\right) \quad (2-20)$$

Note that [Equation 2-20](#) holds separately for each vibrational mode in the system at its resonant frequency. In the frequency domain it is possible to use frequency dependent values of α_{dM} and β_{dK} . For example setting $\alpha_{dM} = 0$ and $\beta_{dK} = 2\zeta/\omega_0$ produces a equivalent viscous damping model at the resonant frequency.

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 2-17](#)). In the frequency domain it is possible to introduce loss factor damping, which has the desired property of frequency independence.

A complication with the Rayleigh damping model is to obtain good values for the damping parameters α_{dM} and β_{dK} . A more physical damping measure is the relative damping, the ratio between actual and critical damping, often expressed as a percentage of the critical damping. Commonly used values of relative damping can be found in the literature.

Using Equation 2-20, this relationship at two frequencies, f_1 and f_2 , with different relative damping, ζ_1 and ζ_2 , results in an equation system that can be solved for α_{dM} and β_{dK} :

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

The damping ratios are then

$$\alpha_{dM} = 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{dK} = \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi(f_2^2 - f_1^2)}$$

Using the same relative damping, $\zeta_1 = \zeta_2$, 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.

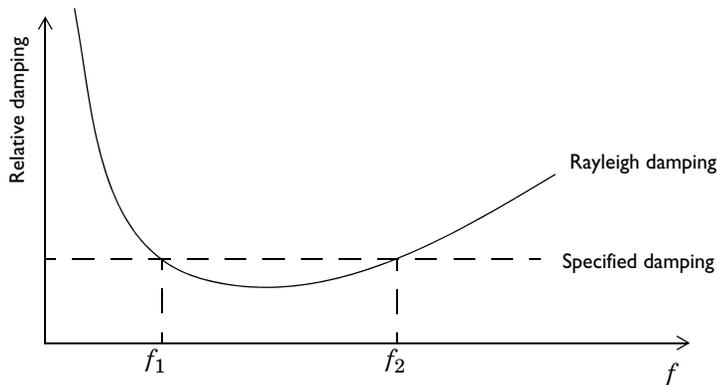


Figure 2-13: An example of Rayleigh damping.

Since the coefficients α_{dM} and β_{dK} should not be negative, the damping ratios are constrained by the respective frequencies as

$$\frac{f_1}{f_2} \leq \frac{\zeta_2}{\zeta_1} \leq \frac{f_2}{f_1}$$

For many applications it is sufficient to leave α_{dM} as zero and to define damping only using the β_{dK} coefficient. Then according to [Equation 2-20](#), 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 β_{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 η_s . But there is no reason that η_s must be scalar. Because the loss factor is a value deduced from true complex-valued material data, it can be represented by a matrix of the same dimensions as the anisotropic stiffness matrix. Especially for orthotropic materials, there should be a set of loss factors of all normal and shear elasticity modulus components, and COMSOL allows all these options, so a more general expression is.

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

For hyperelastic materials the loss information appears as a multiplier in strain energy density, and thus in the second Piola-Kirchhoff stress, S :

$$S = (1 + j\eta_s) \frac{\partial W_s}{\partial E}$$

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 β_{dK} , to the loss factor, η , divided by the excitation frequency:

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

The mass damping factor, α_{dM} , should be set to zero.

If, on the other hand, you would want to use a viscous damping, corresponding to a certain Rayleigh stiffness damping, the conversion to bulk and shear viscosity can be made using the expressions

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

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 η_s can all be frequency dependent.

Piezoelectric Damping

Piezoelectric losses are more complex and include coupling and electrical losses in addition to the material terms. For damping in piezoelectric materials, see [Piezoelectric Losses](#).

For piezoelectric materials, β_{dK} is only used as a multiplier of the structural contribution to the stiffness matrix when building-up the damping matrix as given by [Equation 2-19](#). 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 is 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 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.

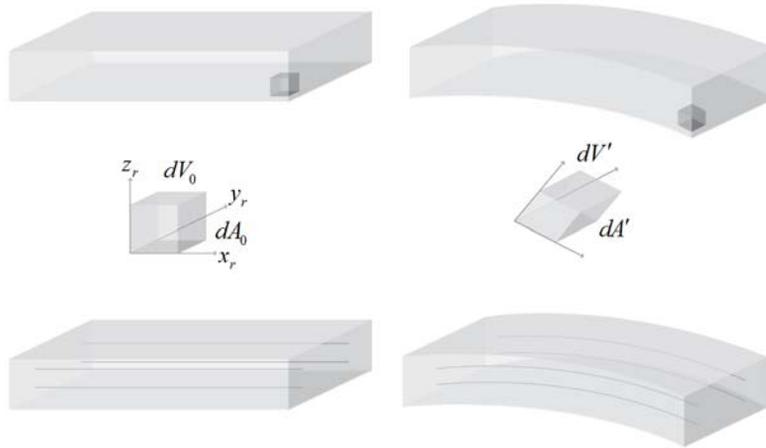


Figure 2-14: An example of the deformation of a beam showing the undeformed state (left) and the deformed state (right) of the beam itself with an element near its tip highlighted (top), of the element (center) and of lines parallel to the x -axis in the undeformed state (bottom).



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

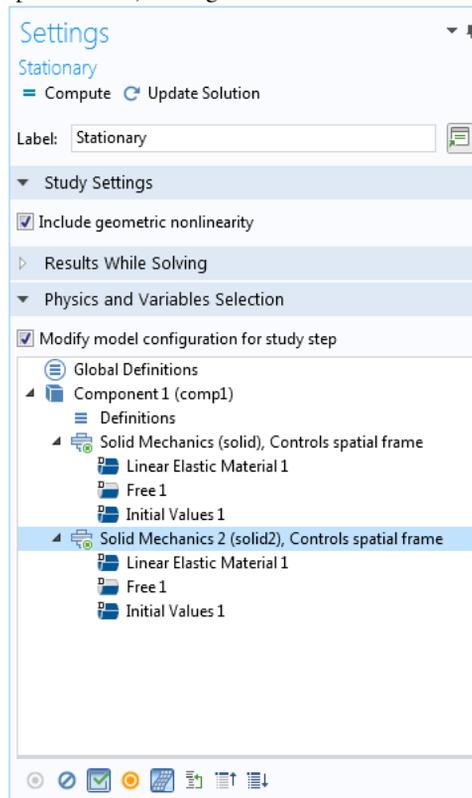
Under geometric nonlinearity, a Solid Mechanics interface (or a similar structural mechanics interface, for example Multibody Dynamics) will assume control over the spatial frame in the domains where it is active. However, the definition of the spatial frame must be unique, and there may be conflicts.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

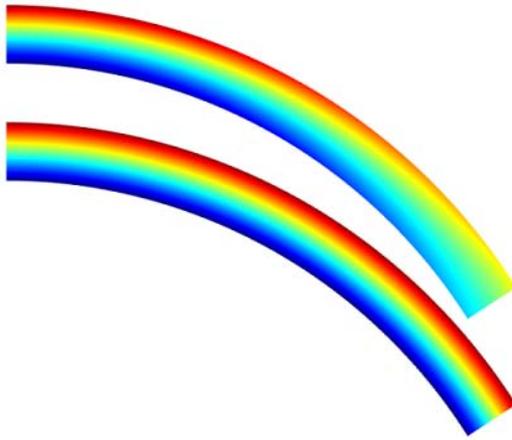


Figure 2-15: xx -components of the Cauchy stress tensor (top) and Second-Piola-Kirchhoff stress tensor (below) for an initially straight beam

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.

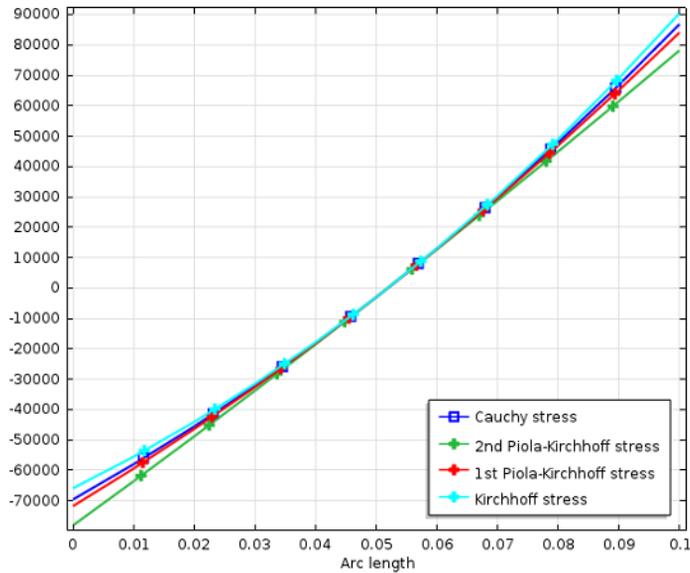


Figure 2-16: Stress distribution across the beam at the constrained end

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 physics features which force this behavior are:

- 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 is not forced by the program.

When you select a geometrically nonlinear study step, the behavior of several features differ 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.

INELASTIC STRESSES AND STRAINS

When inelastic stresses and strains are part of the problem description, you need to make some extra considerations. Such contributions are formally part of the constitutive model, but generate load vector contributions.



There are three **Preset** study types which can be used to set up these two study steps: **Eigenfrequency, Prestressed; Frequency Domain, Prestressed;** and **Linear Buckling**.



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



[Eigenfrequency, Prestressed, Frequency Domain, Prestressed,](#) and [Linear Buckling](#) in the *COMSOL Multiphysics Reference Manual*



- For an example of a general prestressed eigenfrequency analysis see *Bracket — Eigenfrequency Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_eigenfrequency**.
- For an example of an analysis where the stress state is explicitly prescribed, see *Vibrating Membrane*: Application Library path **Structural_Mechanics_Module/Verification_Examples/vibrating_membrane**.

Geometric Nonlinearity for the Piezoelectric Material

PIEZOELECTRIC MATERIALS WITH LARGE DEFORMATIONS

The linear piezoelectric equations as presented in [About Piezoelectric Materials](#) with engineering strains are valid if the model undergoes only relatively small deformations. As soon as the model contains larger displacements or rotations, these equations produce spurious strains that result in an incorrect solution. To overcome this problem, so-called large deformation piezoelectrical equations are required.

The Piezoelectric Material implements the large deformation piezoelectrical equations according to Yang (Ref. 1). Key items of this formulation are:

- The strains are calculated as the Green-Lagrange strains, ε_{ij} :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \sum_k \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) \quad (2-21)$$

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; ε 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 ε_{rS} are the piezoelectric material constants. The expression within parentheses equals the dielectric susceptibility of the solid:

$$\chi = \varepsilon_{rS} - \mathbf{I}$$

The electric displacement field in the material orientation results from the following relation

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

where \mathbf{C} is the right Cauchy-Green tensor

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}$$

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

$$\begin{aligned}
 \mathbf{E} &= \mathbf{F}^{-T} \mathbf{E}_m \\
 \mathbf{P} &= \mathcal{J}^{-1} \mathbf{F} \mathbf{P}_m \\
 \mathbf{D} &= \mathcal{J}^{-1} \mathbf{F} \mathbf{D}_m \\
 \rho_v &= \rho_v \mathcal{J}^{-1}
 \end{aligned}
 \tag{2-22}$$

where \mathbf{F} is the deformation gradient; \mathcal{J} is the determinant of \mathbf{F} ; and ρ_v and ρ_v are the volume charge density in spatial and material coordinates respectively. The deformation gradient is defined as the gradient of the present position of a material point $\mathbf{x} = \mathbf{X} + \mathbf{u}$:

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

Finally, one can rewrite the constitutive equations as

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

DECOUPLED MATERIALS WITH LARGE DEFORMATIONS

The large deformation formulation described in the previous section applies directly to non-piezoelectric materials 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 [Piezoelectric Devices 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

General

Situations where objects come into contact with each other occur frequently in mechanical simulations. Setting up and solving contact problems is sometimes 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. If friction is ignored, 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* and *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.

In this section:

- [Setting up a contact problem](#)
- [Contact Pairs](#)
- [Meshing for Contact Analysis](#)
- [Constraints](#)
- [Settings for Contact Nodes](#)
- [Time-Dependent Contact Analysis](#)
- [Multiphysics Contact](#)
- [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 in the Solid Mechanics and Multibody Dynamics interfaces. To model a mechanical contact problem, you must do the following:

- Add **Contact Pair** nodes under **Definitions**. A contact pair consists of two sets of boundaries, which are called the *source* and *destination* 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.
- 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.
- Select a suitable study type. You can analyze contact using a Stationary or Time Dependent study step.



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

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 in other physics interfaces is always supplied by the structural mechanics interface.

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

INCLUDING FRICTION

In real life, there is always some friction between contacting objects, but this is often ignored. 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 a minor local effect.
- 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.
- 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 stresses in the contact boundaries.
- When a tangential force is necessary to maintain stability and avoid rigid body motions. In many cases, it is however possible to replace the friction by a suitable constraint instead, as long as there are no resultant forces being resisted by such a constraint.
- 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.

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 interesting if you are interested in modeling the 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 rule. You can choose between several different decohesion laws.



The energy dissipated due to decohesion is not accounted for in energy balances or in heat source terms.

SELECTING THE CONTACT ALGORITHM

In COMSOL Multiphysics, there are two possible methods for solving contact problems: the *augmented Lagrangian* method and the *penalty* method.

The default augmented Lagrangian method provides better accuracy, but at a higher computational cost. It requires additional degrees of freedom, and is less stable from the convergence point of view. This method ensures that there will not be any

penetration between the contacting objects in a well converged solution. The contact pressure and friction forces are added as degrees of freedom. The contact pair is *asymmetric* (that is, it is a source/destination pair). The *destination* contact domain is constrained not to penetrate the *source* domain, but not vice versa. The contact condition is evaluated in the integration points on the destination boundary. It is thus possible for a node to have a small penetration into the source boundary, even in a converged state.

In comparison, the penalty method is rather simple and robust. Roughly speaking, it is based on inserting a stiff 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.

The penalty method must be used when modeling adhesion.

You can select the method used for computing the normal direction pressure and the friction forces independently.

Contact Pairs

To decide which boundaries to assign as source and destination in a contact pair consider the following guidelines:

- Make sure that the source boundary stiffness in the normal direction is higher than the destination boundary stiffness. This is especially important if the difference in stiffness is quite large (for example, over ten times larger).
- If one of the boundaries belongs to a part that is rigid, either since it is a rigid domain, or because of the constraints applied, it should be selected as the source boundary.
- 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.

For efficiency, include only those boundaries that can actually come in contact in the destination selection. For the source, it is often a bit more efficient to make it so large that every destination point ‘sees’ a corresponding source point. The source point is obtained by following the normal to the destination until it reaches the source boundary.

FIXED GEOMETRY CONTACT

In some situations, the relative sliding between the contacting boundaries is small. This is often the case for shrink fit simulations or when mounting a component using prestressed bolts. The sliding can be considered as small if it is significantly less than the length of an element edge.

In this case, 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 certain point on the destination boundary will see the same point on the source boundary during the entire simulation. This will make the contact analysis to run faster and convergence to be more stable.

The analysis is still geometrically nonlinear when using this option, and the contact region can still have arbitrarily large displacements and rotations.

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 rules above, you may need to switch them using the **Swap Source and Destination** button in the **Source Boundaries** section of the pair settings.

Meshing for Contact Analysis

Once the source and destination boundaries are chosen, 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 destination at least two times finer than 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. So 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 normals to the contact boundaries change 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.

Constraints

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 causes 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 fully constrained only once 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 solution to the stability 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 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.

Settings for Contact Nodes

PENALTY FACTOR

An important parameter in the settings for the [Contact](#) node is the penalty factor. It is used for both the augmented Lagrangian method and the penalty method, but the interpretation is different:

- In the augmented Lagrangian method, the penalty factor is a numerical parameter, which affects the convergence properties of the algorithm. A higher value gives a faster convergence, but at the price of less stability.
- 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.
- When modeling adhesion, the penalty factor is the stiffness in the normal direction of the adhesive layer.



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.

When running into convergence problems, check the penalty settings. If the iteration process fails in some of the first iterations, lower the penalty parameters. If the model seems to converge but very slowly, consider increasing the maximum value of the penalty parameters.

Increasing the penalty factor can lead to an ill-conditioned Jacobian matrix and convergence problems in the Newton iterations. This is often seen by noting that the damping factor becomes less than 1 for many Newton iterations or that the structure “jumps” into an unphysical state. If this occurs, decrease the penalty factors.

The default values for the penalty factors is based on an “equivalent” Young’s modulus (E_{equiv}) of the material on the destination side. For linear elastic isotropic materials E_{equiv} is equal to the actual Young’s modulus. For other types of materials E_{equiv} is defined by an estimate of a similar stiffness at zero strain. For materials that are user

defined or in other senses nonstandard (for example, anisotropic with large differences in stiffness in different directions), `Eequiv` 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.

Penalty Factor Relaxation

When using the augmented Lagrangian method, having a well tuned penalty factor is an important factor for the performance of the contact iterations.

The default value is selected as a compromise between speed and stability, but with more weight on stability. The strategy is to for each new step (parametric step or time step) start with a softened penalty factor, which is then increased over the first four iterations. The purpose is to stabilize the problem if there are large overclosures in 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.

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** or **Speed**. With **Stability**, relaxation is used in every step. With **Speed**, a constant penalty factor is used all the time, and the value used is also higher than the final value obtained when using **Stability**.
- With **Manual tuning**, you can make adjustments to the magnitude of the penalty factor, and to 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:

- 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. You can then enter a logical expression which, when fulfilled, will force the solver to immediately abandon the iterations and try a smaller step. 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.

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 is included in the gap by using the 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.

When the source or 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 distance. This adjustment can be combined with an offset. The offset is applied to the adjusted value.

The mesh as such is not adjusted, it is only the distance computation which is affected. 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.

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 prohibits the deformation except the contact, 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 never come into contact.

DISCRETIZATION

When using the augmented Lagrangian method it is possible to change the order of the shape functions used for the contact pressure and friction force degrees of freedom. The default is linear shape functions, and this matches the quadratic shape functions used as default for the displacement degrees of freedom in the Solid Mechanics interface. The only situation when you should consider changing the discretization for the contact variables is if you use cubic or higher shape functions for the displacements.

Selecting any other discretization else than **Linear** requires that the solver sequence is modified manually, since the default lumped solver is no longer optimal for the contact pressure update. An ordinary segregated step should then be used.

ADHESION

When using the penalty method, you can specify that the boundaries in the contact pair should stick to each other after coming into contact.

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.

Decohesion

Two boundaries joined by adhesion can separate if you select any of the decohesion laws. You must then specify the maximum strength and the total energy release during decohesion for pure tension and pure shear. In most cases, the decohesion process is a mixture between tension and shear, so you also have to specify the interaction rule.

As long as the stress is below the maximum strength, the adhesive layer retains its original stiffness. Once the displacement in the layer is so large that the maximum strength is exceeded, the material is considered as damaged, and the stiffness is decreased. The same damage factor is applied to both the tensile and the shear stiffness, while the compressive stiffness is unchanged.

Decohesion is an inherently unstable process. The elastic energy in the strained adhesive layer will be released during decohesion. Numerically, the decreasing stress-strain curve manifests itself as a local negative stiffness. Such a problem may be possible to solve, if the surrounding material can absorb the released energy. The numerical stability is closely coupled to the physical stability of the structure.

- If the structure is in 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.
- 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 that is used for post-buckling problems.
- You may have to change the settings in the **Method and Termination** section of the settings for the **Fully Coupled** or **Segregated** nodes in the solver sequence, for example by allowing a larger number of iterations.

- The true unstable failure 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.



- Documentation of the [Contact](#), [Friction](#) and [Adhesion](#) nodes.
- [Adhesion](#) and [Decohesion](#) in the [Structural Mechanics Theory](#) chapter.



For an example showing 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**

Time-Dependent Contact Analysis

The contact formulation is strictly valid only for stationary problems. It is still possible to use contact modeling in a time-dependent analysis, as long as inertial effects are not important in the contact region or you use a sufficiently short time step. In practice, this means that you can solve both *quasi-static problems* and truly dynamic problems, as long as situations with impact are avoided. When in doubt, try to do an *a posteriori* check of conservation of momentum and energy to ensure that the solution is acceptable.

In time-dependent contact models, the penalty method is usually the better choice.

Multiphysics Contact

Multiphysics contact problems often exhibit a high degree of nonlinearity, which leads to convergence problems for 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 each iteration,



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 of the contact area. 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.

Fallback Nodes for Contact

When a **Contact** node has been added, it always contains one subnode, **Free**. This is the default fallback condition on the part of the boundary where contact is not established. By right-clicking the **Contact** node, you can replace it by another 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** fallback node.



- 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 physical 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.

Solver Settings for Contact Analysis

The following solver settings can help to successfully solve contact models:

- 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 then 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 instead of 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 generally used default solver for other problems. Using the same 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 load 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 solver.
- It is important to scale the contact degrees of freedom manually. 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 which 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.
- For some contact problems, it is necessary to let the parametric solver use a more defensive strategy when going to the next parameter step. This can be controlled by setting the value of **Predictor** in the **Parametric** feature to **Constant**.
- When using the augmented Lagrangian method, at least one lumped step will be generated in the segregated solver for each **Contact** node. The number of lumped steps will be increased even more if you select the **Group contact variables in solver per pair** check box in the **Advanced** section of the settings for the **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 convergence problems.
- If the model includes friction, try solving the problem without friction first. When the model seems to work without friction, friction can be added.

- Always solve contact problems which contain 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.
- For models that include decohesion, see also suggestions under [Decohesion](#).

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. Note that if you select **Results while solving** in the **Segregated** feature, 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 (*solid.Tnmax_pair*) and the minimum gap distance (*solid.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 increase the granularity further, by selecting the **Group contact variables in solver per pair** check box in the **Advanced** section of the settings for the **Contact** node.

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 **Friction** nodes will generate a number of degrees of freedom which depend 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. These variables appear in the **Previous Solution** node. In a stationary solver, the path to this node is **Stationary Solver>Parametric>Previous Solution**. In a time-dependent solver it appears directly under **Time-Dependent Solver**.



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** and **Previous Solution** nodes.

In [Table 2-9](#) the dependent variables which can be created by the **Contact** or **Friction** nodes 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 `comp1.solid.Tn_p1`. In the table, it is shown as `Tn`.

TABLE 2-9: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

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

TABLE 2-9: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
cm_old	Previous mapped source coordinates	The location on the source where this point was located at last converged solution	Friction node present	Previous Solution
sliptot_old	Previous accumulated slip	Total slip in this point at last converged solution	Friction: Store accumulated slip selected	Previous Solution
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	Previous Solution
isContact_old	Contact status variable in previous step	Nonzero if the point was in contact at last converged solution	Contact: Add contact status to solver log selected	Previous Solution
isSliding_old	Sliding friction status variable previous step	Nonzero if the point was in a sliding state at last converged solution	Contact: Add contact status to solver log selected. Friction node present.	Previous Solution
isSticking_old	Sticking friction status variable previous step	Nonzero if the point was in a sticking state at last converged solution	Contact: Add contact status to solver log selected. Friction node present.	Previous Solution
Wcnt	Contact energy density	Elastic energy stored in normal contact by the penalty factor.	Contact: Store energy variables selected.	Segregated step

TABLE 2-9: DEPENDENT VARIABLES OCCURRING IN CONTACT ANALYSIS

VARIABLE NAME	DESCRIPTION	EXPLANATION	CREATED	SOLVER NODE
Wfric	Frictional dissipation energy density	Energy dissipated by friction	Contact: Store energy variables selected. Friction node present.	Segregated step
cma_old	Previous adhesive mapped source coordinates	Position on the source where this point was located when adhesion was detected.	Adhesion node present	Previous Solution
u_max_old	Maximum adhesion displacement at previous step	Maximum norm of adhesive layer displacement reached until previous step. Also used as indication of adhesion activation.	Adhesion node present	Previous Solution

Important Contact Variables

In this section you will find a summary of variables created in the contact feature which can be useful in postprocessing.

TABLE 2-10: IMPORTANT CONTACT VARIABLES

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.gap_p1	Gap distance including offsets, contact pair p1	Contact	
solid.gapmin_p1	Minimum gap distance, contact pair p1	Contact	
solid.cnt1.Tn	Contact pressure	Contact	
solid.Tnmax_p1	Maximum contact pressure, contact pair p1	Contact	
solid.cnt1.fric1.Ttnorm	Friction force norm	Friction	

TABLE 2-10: IMPORTANT CONTACT VARIABLES

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.cnt1. fric1.Ttx	Friction force, x component	Friction	
solid.Wcnt_tot	Total contact energy	Contact	Integration over all contact boundaries
solid.Wfric_tot	Total frictional dissipation energy	Friction	Integration over all contact boundaries

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, you add the **Activation** subnode. You then select the domains that you want to subject to **Activation**. The **Activation expression** input 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 the default values 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.

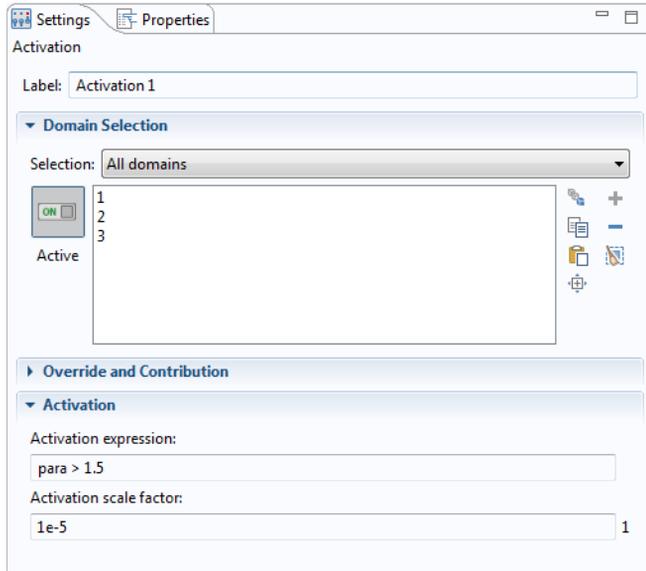


Figure 2-17: Settings for the Activation subfeature.

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
```

TABLE 2-11: 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

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

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 which actually are active, see [Figure 2-18](#).

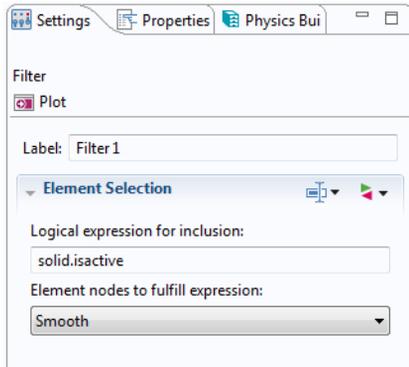


Figure 2-18: Filter settings to only display active material.

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 internal 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 η 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 Interface](#) included with this module has a predefined one-way coupling 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.

By default, COMSOL Multiphysics takes advantage of the one-way coupling 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 *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 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](#)

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 coupling 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 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 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 has a temperature dependence. In general you have to supply the temperature in all the **Model Inputs** sections. This can be done either by explicitly giving a temperature or by selecting a temperature variable from another physics interface.



If there is a **Temperature Coupling** node under the **Multiphysics** branch, then you cannot change the temperature input under Model Inputs. The temperature is automatically taken from the connected Heat Transfer in Solids interface.

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 density*. Conceptually, you can consider this as the temperature at which the domain has the size in which it is drawn. In practice, the choice of reference temperature is seldom an issue, unless your application requires extreme precision. The density of a solid material has a rather slow variation with temperature, so in most cases it is sufficient to use room temperature as reference.

If any material in the model has a temperature dependent mass density, the **Volume reference temperature** list will appear in the **Model Input** section of the material settings. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.

All effects of volume change with temperature are incorporated through the thermal expansion effects.

Thermal Expansion

As the temperature changes, most materials react by a change of volume. For a constrained structure, the stresses that evolve even with moderate temperature changes can be considerable. The volume change can be represented a *thermal strain* ϵ_{th} , which produces stress-free deformations. For a linear elastic material, the constitutive law is

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

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 [The Fluid-Solid Interaction Interface](#) and [Coupled Interfaces](#) (described in the *COMSOL Multiphysics Reference Manual*).
- When the temperature distribution is computed by another physics interface (often Heat Transfer in Solids), it is a good practice to use a discretization which in the heat transfer interface is one order lower than what is used in the structural mechanics interface. The thermal strains are proportional to the temperature, while the strains in the structural problem are computed as derivatives of the displacements, so the approximations will then be consistent. This is automatically taken care of when you use [The Thermal Stress Interface](#) or [The Joule Heating and Thermal Expansion Interface](#).

TEMPERATURE DEPENDENCE OF THE THERMAL EXPANSION COEFFICIENT

When performing an analysis over a larger range of temperatures, you often need to take into account 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

$$\frac{dL}{L} = \alpha_t(T)dT \quad (2-23)$$

where α_t is the tangential thermal expansion coefficient. This form, which is the thermodynamic definition, is conceptually simple, because α_t is uniquely defined at each temperature. It is, however, less convenient to use in practice because an integration is required for determining the actual change in length for a finite temperature difference.

The secant formulation, which is the default in COMSOL Multiphysics, is often used in engineering:

$$\frac{\Delta L}{L_0} = \alpha(T)\Delta T$$

In the secant formulation, the actual values of α will however depend on the choice of reference temperature, T_{ref} , at which the material has the reference length L_0 :

$$\frac{\Delta L}{L_0} = \alpha(T, T_{\text{ref}})(T - T_{\text{ref}})$$

Converting from Tangent Form to Secant Form

Equation 2-23 can be integrated, giving

$$\ln\left(\frac{L}{L_0}\right) = \int_{T_{\text{ref}}}^T \alpha_t(\tau)d\tau \quad (2-24)$$

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

$$\alpha(T, T_{\text{ref}}) = \frac{I(T, T_{\text{ref}})}{(T - T_{\text{ref}})} \quad (2-25)$$

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 2-24 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 2-23 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 $\langle\text{phys}\rangle.T_{\text{ref}}$ and the current temperature $\langle\text{phys}\rangle.T$, where $\langle\text{phys}\rangle$ 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 $\langle\text{phys}\rangle.T$.

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:

$$L(T) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \quad (2-26)$$

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

$$L(T_{\text{ref}}) = [1 + \alpha_m(T_{\text{ref}})(T_{\text{ref}} - T_m)]L(T_m) \quad (2-27)$$

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 2-26](#) but with T_m replaced by T_{ref} .

$$L(T) = [1 + \alpha_r(T)(T - (T_{\text{ref}}))]L(T_{\text{ref}}) \quad (2-28)$$

Here $\alpha_r(T)$ is the redefined thermal expansion coefficient, based on T_{ref} . It can be derived from the relations above. Using [Equation 2-26](#) and [Equation 2-28](#) there are two ways of writing the current length $L(T)$, so that

$$[1 + \alpha_r(T)(T - (T_{\text{ref}}))]L((T_{\text{ref}})) = [1 + \alpha_m(T)(T - T_m)]L(T_m) \quad (2-29)$$

Equation 2-27 makes it possible to eliminate $L(T_{\text{ref}})$ and $L(T_m)$ from Equation 2-29, and after some algebra $\alpha_r(T)$ can then be written as

$$\alpha_r(T) = \frac{\alpha_m(T) + (T_{\text{ref}} - T_m) \frac{\alpha_m(T) - \alpha_m(T_{\text{ref}})}{T - T_{\text{ref}}}}{1 + \alpha_m(T_{\text{ref}})(T_{\text{ref}} - T_m)} \quad (2-30)$$

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 2-30, but with a small modification to avoid problems if $T = T_{\text{ref}}$.

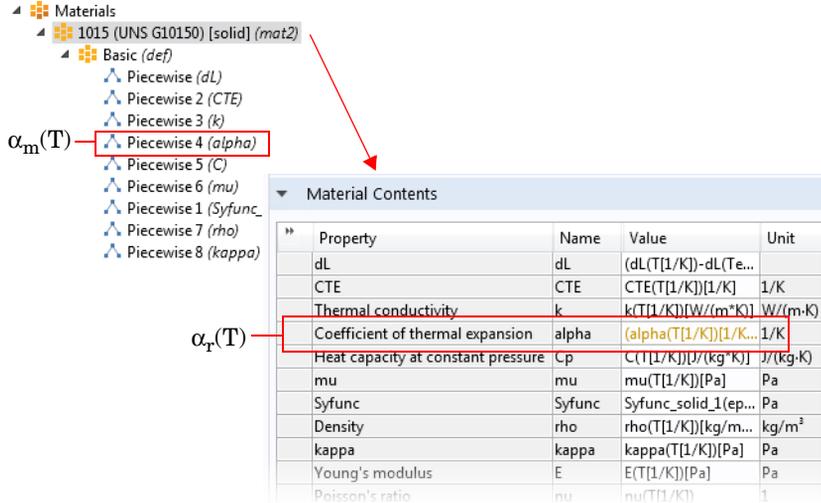


Figure 2-19: An example in COMSOL Multiphysics showing the Materials branch and where to find the temperature-dependent thermal expansion coefficient.

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. This is particularly

important if the measured data used the Fahrenheit scale (denoted by [degF] in COMSOL Multiphysics).

- 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 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 which 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 taken into account.

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 is 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 the 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 which 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

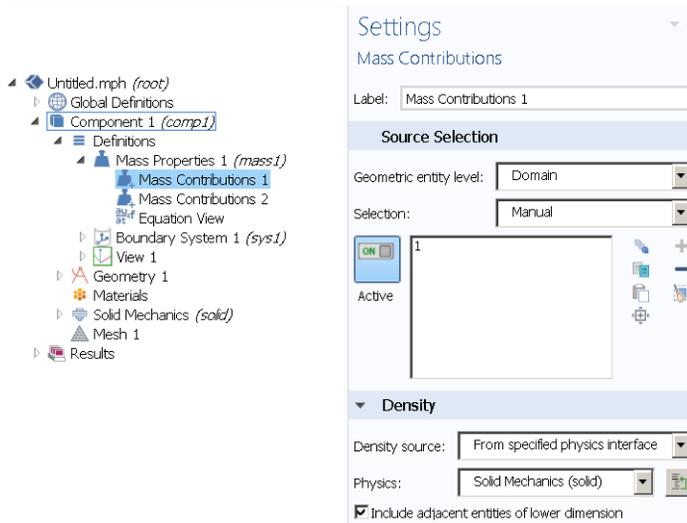
Sometimes the structural deformations are so large that object may come into contact with each other with the fluid being squeezed in-between. Modeling contact together with FSI requires some special considerations. The mesh in the fluid domain may deform, but the topology remains the same — a fluid 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.

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 can find in the equation view under the corresponding **Mass Properties** node.

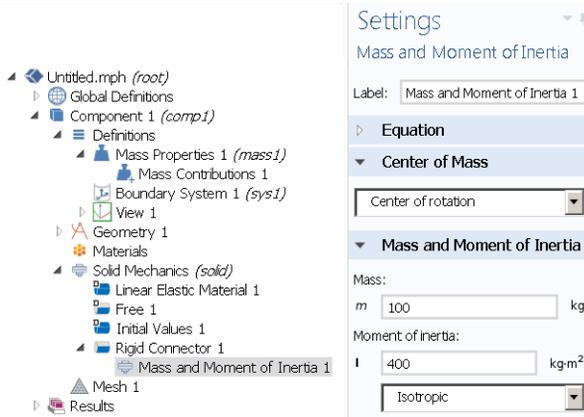
Variables			
Name	Expression	Unit	Description
mass1.volume	mass1.intDim3(root.round(mass1.con...	m ³	Volume
mass1.mass	mass1.intDim0(mass1.rho_pnt)+mass...	kg	Mass
mass1.CMX	(mass1.intDim0(mass1.rho_pnt*X)+...	m	Center of mass, X component
mass1.CMY	(mass1.intDim0(mass1.rho_pnt*Y)+...	m	Center of mass,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

Besides for postprocessing purposes, you can also use these variables in any user-defined expressions, user inputs, and in optimization criteria.

Structural mechanics interfaces contribute to the mass properties in several ways:

- All material models, including Rigid 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 will contribute both with mass and with the specified mass moment of inertia. It is possible to suppress the contribution from an **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 micromechanics material properties cannot be directly used in the macroscopic scale in which the structure is to be analyzed. In such a case, you typically want to use a homogeneous material model with appropriate average 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 which contains a representation of the material in the microscale.

If the material is truly periodic, the RVE typically is 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 macro 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 mesh matches.

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



Micromechanical Model of a Composite: Application Library path

**Structural_Mechanics_Module/Tutorials/
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 internal 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.

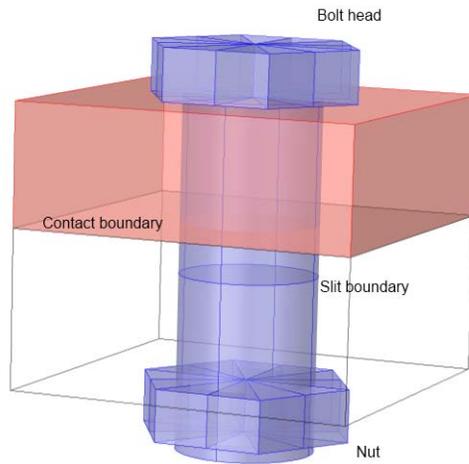


Figure 2-20: Example of a bolted joint.

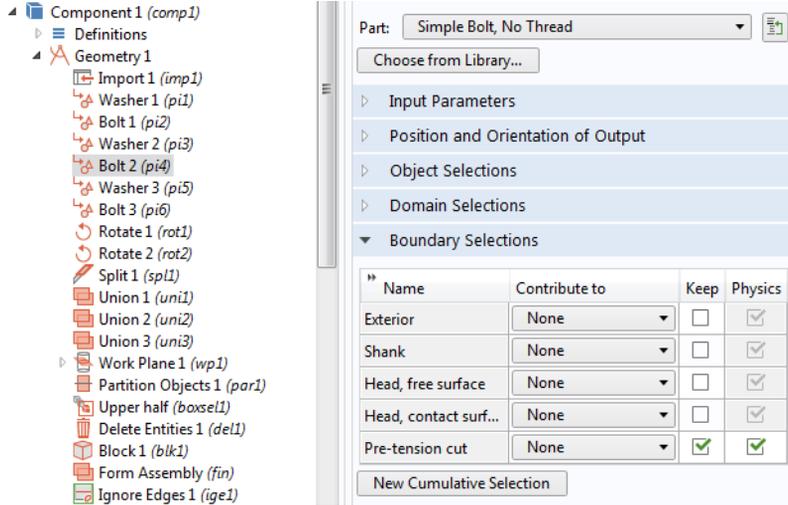


Figure 2-21: Getting access to the slit boundary selection for a bolt from the Part Libraries.

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 2-12: 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 parametrized bolt geometries which 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 control of search distance** 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.

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 relative damping. The response value is provided as a function of the natural frequency of the oscillator.

If the dynamic structural response is viewed as a modal 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 an approach will in most cases result in 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

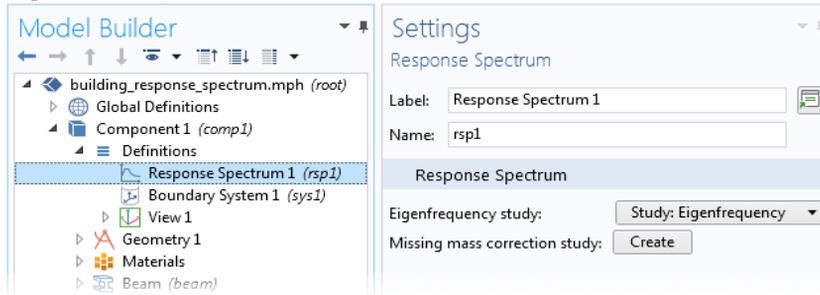
The response spectrum analysis is not a study type. The computations are performed during result evaluation. You enter the methods and parameters in the settings for the **Response Spectrum 2D** or **Response Spectrum 3D** data sets. These data sets require the following:

- 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 data set 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 which 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

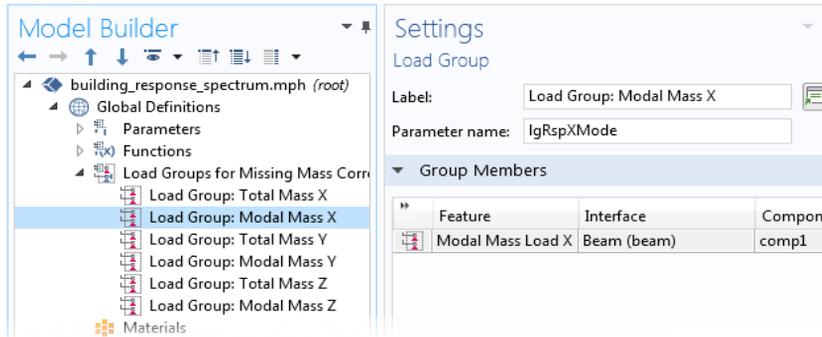
You create a missing mass correction study from the **Response Spectrum** node by clicking the **Missing mass correction study: Create** button.



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:

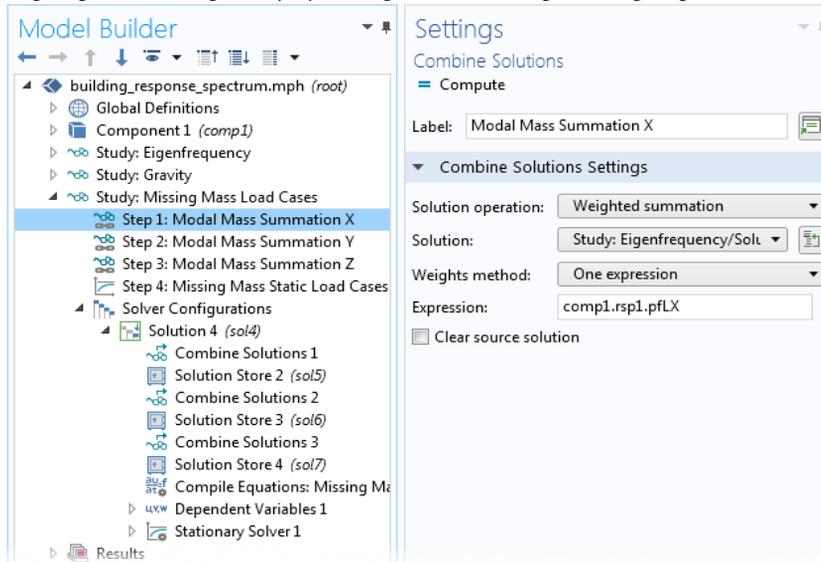
- A set of load group nodes are created under **Global Definitions**. There are two load groups for each spatial direction. The load group nodes are placed under a common group named **Load Groups for Missing Mass Correction**.



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** drop-down 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.



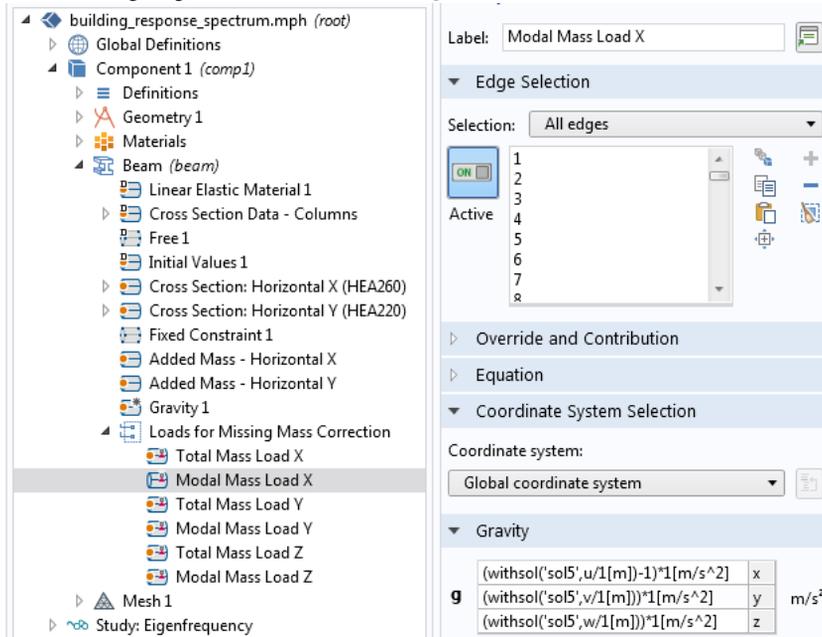
If the physics interface has loads other than those automatically generated for missing mass correction, you need to make sure that those loads are not solved for in the **Missing Mass Static Load Cases** study step. There are several ways of doing that. You can, for example:



- 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 DATA SETS

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 respectively. Those spectra act in two orthogonal directions which by default coincide with the global X and Y directions. By giving a non-zero value to **Primary axis rotation**, you can make the primary spectrum act in an arbitrary direction in the X-Y 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 Guide*.
- [Combine Solutions](#) and [Updating a Solution](#) in the *COMSOL Multiphysics Reference Guide*.
- The settings for the special data sets are described in [Response Spectrum 2D](#) and [Response Spectrum 3D](#) in the *COMSOL Multiphysics Reference Guide*.



- *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 dimensioning 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 X-Y 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 R-Z 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.

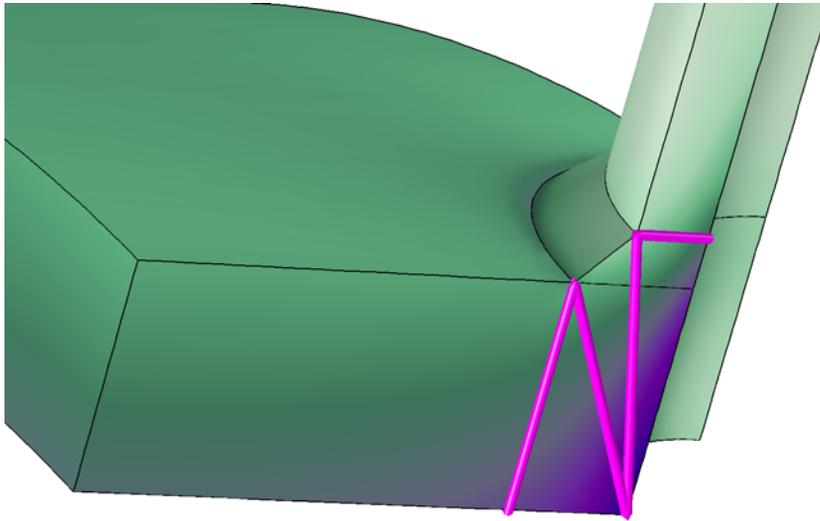


Figure 2-22: Four stress classification lines in a transition region at a pressure vessel nozzle.

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` which 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 data sets and an extra default plot are generated.

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

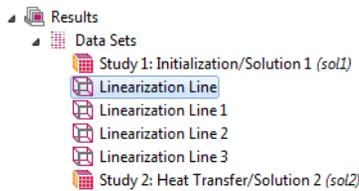


Figure 2-23: Generated data sets in a model with four SCL.

The default plot contains graphs for the 22 component of the actual stress, the membrane stress, and the linearized stress. The first **Linearization Line** data set is selected. By changing edge date set in the plot group, you can easily move between the different stress classification lines.

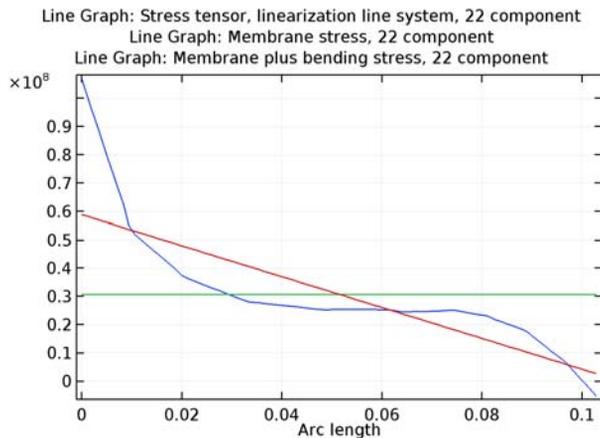


Figure 2-24: Default plot along a stress classification line.

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 makes 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 2-13: VARIABLE FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Sllj	Stress tensor in local coordinate system	σ_{ij}	ij = 11, 12, 13, 22, 23, 33	Physics, Feature
Smij	Membrane stress	$\sigma_{m,ij}$	ij = 11, 12, 13, 22, 23, 33	Physics, Feature
Sbmaxij	Maximum bending stress	$\sigma_{b(max),ij}$	ij = 11, 12, 13, 22, 23, 33	Feature
Sbij	Bending Stress	$\sigma_{b,ij}$	ij = 11, 12, 13, 22, 23, 33	Physics, Feature
Smbij	Membrane + bending stress	$\sigma_{mb,ij}$	ij = 11, 12, 13, 22, 23, 33	Physics, Feature
Spsij	Peak stress, starting point	$\sigma_{p(start),ij}$	ij = 11, 12, 13, 22, 23, 33	Feature
Speij	Peak stress, starting point	$\sigma_{p(end),ij}$	ij = 11, 12, 13, 22, 23, 33	Feature
Slm	Stress intensity, membrane	σ_{int}		Physics, Feature
Slmbs	Stress intensity, membrane + bending, starting point	σ_{int}		Feature
Slbme	Stress intensity, membrane + bending, endpoint	σ_{int}		Feature

TABLE 2-13: VARIABLE FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Slmb	Stress intensity, membrane + bending,		max(Slmb _s , Slmb _e)	Physics, Feature
N _{ij}	Local in-plane force	N_{ij}	$ij = 22, 23, 33$	Feature
M _{ij}	Local bending moment	M_{ij}	$ij = 22, 23, 33$	Feature
Q _i	Local out-of-plane shear force		$i = 2, 3$	Feature
lengthtot	Length of SCL	L		Feature
arclength	Coordinate along SCL	X_1		Feature



[Stress Linearization](#) in the Structural Mechanics Theory Chapter.

Solver Settings

COMSOL Multiphysics includes many solvers and solver settings. To make it easier to use a suitable solver and its associated solver parameters, the physics interfaces have different default settings based on the study type and features used. In some situations, the default settings may need to be changed. This section helps you to select a solver and its settings to solve structural mechanics and related multiphysics problems.

In this section:

- [Symmetric Matrices](#)
- [Selecting Iterative Solvers](#)
- [Specifying Tolerances and Scaling for the 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 the MUMPS direct solver in both 2D and 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/eigenvalue studies, use the default direct solver (MUMPS).



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 effective 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](#)
-

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.

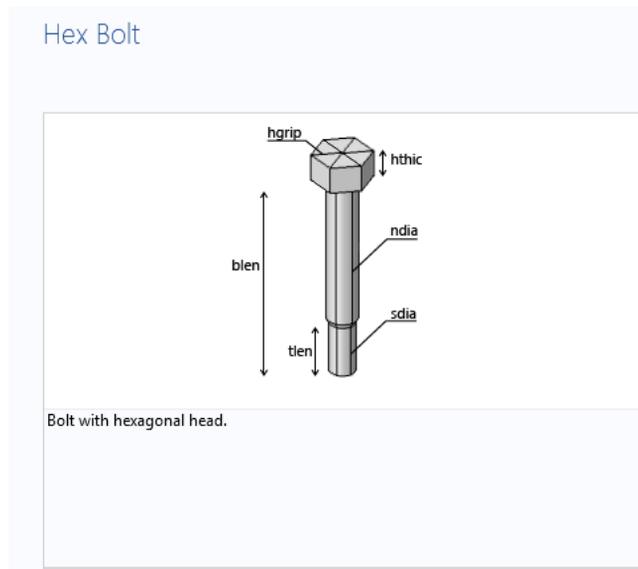
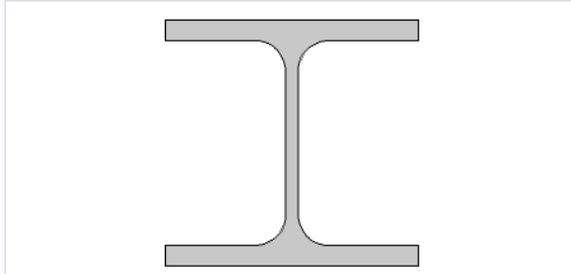


Figure 2-25: Example of a bolt part.

- 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

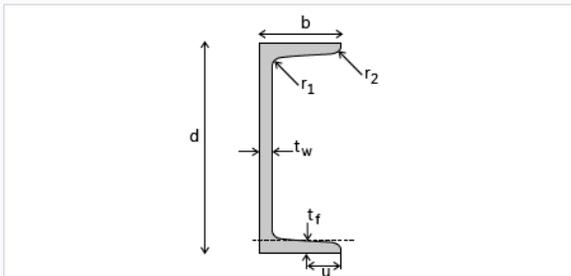


European wide flange beams.

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

Figure 2-26: Example of a standard beam cross section.

Generic C-beam



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

Figure 2-27: 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](#)
- [Material Models](#)
- [Formulation of the Equilibrium Equations](#)
- [Study types](#)
- [Damping](#)
- [Loads and Boundary Conditions](#)
- [Stress Linearization](#)
- [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 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 σ and the strain tensor ε are second-order tensors, while the constitutive tensor C is a fourth-order tensor. The ':' symbol means a contraction over two indices. In a notation where the indices are shown, the same equation would read

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

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

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

In a few cases, non-orthonormal 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 , φ , and z in axisymmetric components).

Continuum mechanics theory also makes use of a second set of coordinates, known as *material* (or *reference*) coordinates. These are normally denoted by uppercase variables X , Y , and Z (or R , Φ , and Z) and are used to label material particles. Any material particle is uniquely identified by its position in some given initial or reference configuration. As long as the solid stays in this configuration, material and spatial coordinates of every particle coincide and displacements are zero by definition.

More information can be found in [About Frames](#) chapter in the *COMSOL Multiphysics Reference Manual*.

When the solid object deforms due to external or internal forces and constraints, each material particle keeps its material coordinates \mathbf{X} (bold font is used to denote coordinate vectors), while its spatial coordinates change with time and applied forces such that it follows a path

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

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

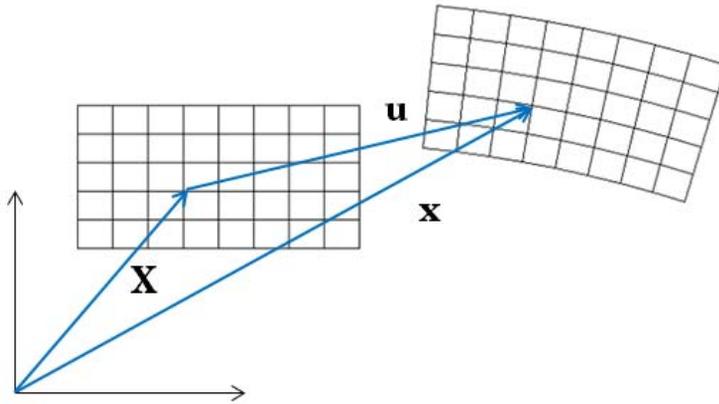
$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$$

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.

$$\nabla = \nabla_X = \left[\frac{\partial}{\partial X} \quad \frac{\partial}{\partial Y} \quad \frac{\partial}{\partial Z} \right]$$

The gradient of the displacement, which occurs frequently in the following theory, is always computed with respect to material coordinates. In 3D:

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & \frac{\partial w}{\partial Z} \end{bmatrix}$$

The *deformation gradient* tensor \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

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

Here, ρ_0 is the initial density and ρ 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 $J = 1$. Also, an incompressible material is represented by $J = 1$. These are called *isochoric* processes.

The determinant of the deformation gradient tensor is always positive (since a negative mass density is unphysical). The relation $\rho = \rho_0/J$ implies that for $J < 1$ there is compression, and for $J > 1$ there is expansion. Since $J > 0$, the deformation gradient \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 ρ_0 , and not the “current” density ρ . If you are interested in finding the density of the deformed material (the density in the spatial frame), you can compute it using the expression `solid.rho/solid.J`.

Deformation Measures

Since the deformation tensor \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 upper case index refers to the material frame, and a lower case 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, λ_1 , λ_2 , and λ_3 . The stretch of a line element with initial length L_0 and current length L is

$$\lambda = \frac{L}{L_0} = 1 + \epsilon_{\text{eng}}$$

where ϵ_{eng} is the engineering strain. The three principal stretches act along three orthogonal directions. In the coordinate system defined by these principal directions, the 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

$$\epsilon = \frac{1}{2}(C - I) = \frac{1}{2}(F^T F - I)$$

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 (ε_{xy} , ε_{yz} , ε_{xz}) is used.



In the documentation, the symbol ε 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 ε consists of both normal and shear strain components:

$$\varepsilon = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix}$$

The strain-displacement relationships for the axial symmetry case for small displacements are

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

A general description of the axially symmetric case is given in [Axial Symmetry and Deformation](#).

Invariants of Strain

PRINCIPAL STRAINS

The principal strains are the eigenvalues of the strain tensor (ε), computed from the eigenvalue equation

$$(\varepsilon - \varepsilon_p I) \mathbf{v}_p = 0$$

The three principal strains are sorted so that

$$\epsilon_{p1} \geq \epsilon_{p2} \geq \epsilon_{p3}$$

This sorting is true also for the 2D cases. The corresponding vectors in the principal directions, \mathbf{v}_{pi} , are orthonormal.



- The internal variables for the principal strains are named `solid.ep1`, `solid.ep2`, and `solid.ep3` respectively.
- 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`, respectively. The elastic principal stretches are named `solid.stchelp1`, `solid.stchelp2`, and `solid.stchelp3`, respectively.

The different invariants of the strain tensor form are useful for constitutive modeling and result interpretation. The three fundamental invariants for any tensor are

$$I_1(\epsilon) = \text{trace}(\epsilon)$$

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

$$I_3(\epsilon) = \det(\epsilon)$$

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

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



The internal variable for the volumetric strain is `solid.evol`.

In terms of the principal strains, the stress 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 stresses are the roots of the *characteristic equation* (Cayley–Hamilton theorem)

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

STRAIN RATE AND SPIN

The spatial velocity gradient is defined in components as

$$L_{kl} = \frac{\partial}{\partial x_l} v_k(\mathbf{x}, t)$$

where $v_k(\mathbf{x}, t)$ is the spatial velocity field. It can be shown that L can be computed in terms of the deformation gradient as

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

where the material time derivative is used.

The velocity gradient can be decomposed into symmetric and skew-symmetric parts

$$L = L_d + L_w$$

where

$$L_d = \frac{1}{2}(L + L^T)$$

is called the *rate of strain tensor*, and

$$L_w = \frac{1}{2}(L - L^T)$$

is called the *spin tensor*. Both tensors are defined on the spatial frame.

It can be shown that the material time derivative of the Green-Lagrange strain tensor can be related to the rate of strain tensor as

$$\frac{d\varepsilon}{dt} = F^T L_d F$$

The spin tensor $L_w(\mathbf{x}, t)$ accounts for an instantaneous local rigid-body rotation about an axis passing through the point x .



Components of both L_d and L_w are available as results and analysis variables under the Solid Mechanics interface.

- The components of the rate of strain tensor are named `solid.Ldx`, `solid.Ldxy`, and so on.
- The components of the spin tensor are named `solid.Lwx`, `solid.Lwxy`, and so on.

Inelastic Strain Contributions

Many of the material models in COMSOL Multiphysics will compute a stress based on an *elastic strain*. The elastic strain tensor is obtained after removing any inelastic deformation contribution from the total deformation from the displacements. There are several possible inelastic strain contributions:

- Initial strain, ϵ_0
- External strain, ϵ_{ex}
- Thermal strain, ϵ_{th}
- Hygroscopic strain, ϵ_{hs}
- Plastic strain, ϵ_{pl}
- Creep strain, ϵ_{cr}
- Viscoplastic strain, ϵ_{vp}

ADDITIVE DECOMPOSITION

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

$$\epsilon_e = \epsilon - \epsilon_{inel}$$

where

$$\epsilon_{inel} = \epsilon_0 + \epsilon_{ex} + \epsilon_{th} + \epsilon_{hs} + \epsilon_{pl} + \epsilon_{cr} + \epsilon_{vp}$$

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}_{inel} tensor, however, describes a mapping from the material frame to an intermediate frame, and the \mathbf{F}_{el} tensor describes a mapping from the intermediate frame to the spatial frame.

When the inelastic deformation gradient tensor is known, the elastic deformation gradient tensor is computed as

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

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}_{el} .

$$\mathbf{C}_{\text{el}} = \mathbf{F}_{\text{el}}^T\mathbf{F}_{\text{el}}$$

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

$$\boldsymbol{\varepsilon}_{\text{el}} = \frac{1}{2}(\mathbf{C}_{\text{el}} - \mathbf{I})$$

The inelastic deformation tensor \mathbf{F}_{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 \mathbf{F}_{inel} .

$$\mathbf{F}_{\text{inel}} = \mathbf{F}_1 \mathbf{F}_2 \mathbf{F}_3 \mathbf{F}_4 \dots$$

where \mathbf{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

$$\det(\mathbf{F}) = \det(\mathbf{F}_{\text{el}})\det(\mathbf{F}_{\text{inel}}) \text{ or } J = J_{\text{el}}J_{\text{inel}}$$



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 nonunit 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

$$I_1(A) = \text{trace}[A_i^i] = \sum_i A_i^i = A_{11} + A_{22} \frac{1}{r^2} + A_{33}$$

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

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

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

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

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

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

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

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

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 , i.e.

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

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 ϵ_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, ω can be computed by an eigenfrequency analysis for the 2D cross section with all three displacement components taken as dependent variables. As a result, one obtains

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

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 ω) can be taken as a parameter. Then, the solution can be computed via eigenvalue analysis with respect to the wave number k_Z using the 2D cross section geometry. Hence,

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

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 ϕ . 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 σ defined as force/deformed area in fixed spatial directions not following the body. This is a symmetric tensor.
- First Piola-Kirchhoff stress P . The forces in the spatial directions are related to the area in the original (material) frame. This is an unsymmetric two-point tensor.
- Second Piola-Kirchhoff stress S . Both force and area are represented in the material configuration. For small strains the values are the same as Cauchy stress tensor but the directions are rotating with the body. This is a symmetric tensor.

The stresses relate to each other as

$$S = F^{-1}P$$
$$\sigma = J^{-1}PF^T = J^{-1}FSF^T$$

In a geometrically linear analysis, the distinction between the stress measures disappear, and they all converge to the same values.



In the documentation, the symbol σ is used to denote not only Cauchy stress, but stress in general. The symbols P and S are used whenever it is necessary to make a distinction. In geometrically nonlinear analysis, the stress should in general be interpreted as second Piola-Kirchhoff stress.



- The internal variables for the Cauchy stresses are named `solid.sx`, `solid.sxy`, and so on.
- The internal variables for the first Piola-Kirchhoff stresses are named `solid.PxX`, `solid.PxY`, and so on.
- The internal variables for the second Piola-Kirchhoff stresses are named `solid.SX`, `solid.SXY`, and so on.

SIGN CONVENTIONS

A positive normal stress in COMSOL Multiphysics acts in tension. This is the most widely used definition in general physics and engineering.

Within the field of geomechanics it is however common to let compressive stresses be positive, since compression is almost always dominant in that field of science. When working with the material models intended for soils and rocks, you must be aware of that the “positive in tension” convention is used also there in order to maintain consistency within the software.

Specifically, the ordering of [Principal Stresses](#) is such that $\sigma_{p1} \geq \sigma_{p2} \geq \sigma_{p3}$ (including signs). In geotechnical applications the dominant principal stresses will usually be compressive, so the third principal stress will 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` respectively.
- 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* θ .

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

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 effective (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 π -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 (σ_{p1} , σ_{p2} , and σ_{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 all out-of-plane shear components of both stress 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 in initial strain ϵ_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*. So 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

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

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

Axial Symmetry and Stresses

The physical stress components are defined on the global coordinate system:

$$\sigma_r = \sigma^{11}$$

$$\sigma_\varphi = r^2 \sigma^{22}$$

$$\sigma_z = \sigma^{33}$$

$$\sigma_{rz} = \sigma^{13}$$

The first invariant of the stress tensor is

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

Equation of Motion

The first Piola-Kirchhoff stress \mathbf{P} is calculated from the second Piola-Kirchhoff stress as $\mathbf{P} = \mathbf{F}\mathbf{S}$. 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:

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \mathbf{F}_{\mathbf{V}} - \nabla_{\mathbf{X}} \cdot \mathbf{P}^T \quad (3-10)$$

where the density corresponds to the material density in the initial undeformed state, the volume force vector $\mathbf{F}_{\mathbf{V}}$ 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 non symmetric 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, σ , can be calculated as

$$\sigma = J^{-1}PF^T = J^{-1}FSF^T$$

The Cauchy stress is a true stress that relates forces in the present configuration (spatial frame) to areas in the present configuration, and it is a symmetric tensor.

Equation 3-10 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 (effective) stress

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

Material Models

There are many material models available for structural analysis in COMSOL Multiphysics. In this section, 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
 - Generalized Maxwell Model
 - Standard Linear Solid Model
 - Kelvin-Voigt Model

- 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
 - User Defined
 - Mullins Effect
- Elastoplastic Material Models
 - von Mises Criterion
 - Tresca Criterion
 - Shima-Oyane Criterion
 - Gurson Criterion
 - Gurson-Tvergaard-Needleman Criterion
 - Fleck-Kuhn-McMeeking Criterion
 - FKM-GTN Criterion
 - Mohr-Coulomb Criterion
 - Drucker-Prager Criterion
 - Matsuoka-Nakai Criterion
 - Lade-Duncan Criterion
 - Hill Orthotropic Plasticity

- Failure Criteria for Concrete, Rocks, and Other Brittle Material
 - Bresler-Pister Criterion
 - Willam-Warnke Criterion
 - Ottosen Criterion
 - Original Hoek-Brown Criterion
 - Generalized Hoek-Brown Criterion
- Elastoplastic Soil Models
- Creep and Viscoplasticity
 - Norton Law (Power law)
 - 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
- 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:

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

where \mathbf{C} is the 4th order *elasticity tensor*, “:” stands for the double-dot tensor product (or double contraction). The elastic strain $\boldsymbol{\varepsilon}_{\text{el}}$ 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

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

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

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

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	$\nu =$	ν	$\frac{E}{2G} - 1$	$\frac{1}{2} \left(1 - \frac{3G}{3K+G} \right)$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	$K =$	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	K	$\lambda + \frac{2\mu}{3}$
Shear modulus	$G =$	$\frac{E}{2(1+\nu)}$	G	G	μ
Lamé parameter λ	$\lambda =$	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K - \frac{2G}{3}$	λ
Lamé parameter μ	$\mu =$	$\frac{E}{2(1+\nu)}$	G	G	μ
Pressure-wave speed	$c_p =$	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\frac{K+4G/3}{\rho}}$	$\sqrt{\frac{\lambda+2\mu}{\rho}}$
Shear-wave speed	$c_s =$	$\sqrt{\frac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

According to [Table 3-1](#), the elasticity matrix D for isotropic materials is written in terms of Lamé parameters λ and μ ,

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

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

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

ORTHOTROPIC AND ANISOTROPIC MATERIALS

There are two different ways to represent orthotropic or anisotropic data. The **Standard (XX, YY, ZZ, XY, YZ, XZ) 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 (XX, YY, ZZ, XY, YZ, XZ) Material data ordering**, the elasticity coefficients can be entered following the **Voigt** notation. In the **Voigt (XX, YY, ZZ, YZ, XZ, XY) 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 material in the **Standard (XX, YY, ZZ, XY, YZ, XZ)** **Material data ordering** has the following structure:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ D_{12} & D_{22} & D_{23} & 0 & 0 & 0 \\ D_{13} & D_{23} & D_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & D_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{bmatrix}$$

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 , ν_{xy} , ν_{yz} , ν_{xz} , G_{xy} , G_{yz} , and G_{xz} are supplied in designated fields in the physics interface. COMSOL Multiphysics deduces the remaining components— ν_{yx} , ν_{zx} , and ν_{zy} —using the fact that the matrices D and D^{-1} are symmetric. The *compliance matrix* has the following form:

$$D^{-1} = \begin{bmatrix} \frac{1}{E_x} & \frac{\nu_{yx}}{E_y} & \frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ \frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & \frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ \frac{\nu_{xz}}{E_x} & \frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} \end{bmatrix}$$



The values of ν_{xy} and ν_{yx} are different for an orthotropic material. For a certain set of given material data, you must make sure that the definition of the indexes is consistent with the definition used in COMSOL Multiphysics.

The elasticity matrix in the **Voigt (XX, YY, ZZ, YZ, XZ, XY) Material data ordering** changes the sorting of the last three elements in the elasticity matrix:

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

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 (XX, YY, ZZ, XY, YZ, XZ)** or **Voigt (XX, YY, ZZ, YZ, XZ, XY) Material data ordering**.

AXIAL SYMMETRY

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

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

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

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

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

$$C_{ijkl}^{\text{phys}}$$

and the corresponding tensor components are computed internally according to:

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

For an isotropic material:

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

where λ and μ are the first and second Lamé elastic parameters and g is the metric tensor.

For a hyperelastic material, the second Piola-Kirchhoff stress tensor is computed as

$$S = 2 \frac{\partial W_s}{\partial C}$$

which is computed as the contravariant components of the stress in the local coordinate system:

$$S^{ij} = 2 \frac{\partial W_s}{\partial C_{ij}}$$

The energy variation is computed as

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

which can be also written as

$$S_r \text{test}(\varepsilon_r) + S_\varphi \text{test}(\varepsilon_\varphi) + S_z \text{test}(\varepsilon_z) + 2S_{rz} \text{test}(\varepsilon_{rz})$$

ENTROPY AND THERMOELASTICITY

The free energy for the linear thermoelastic material can be written as

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

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

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

and the entropy per unit volume can be calculated as

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

where T_0 is a reference temperature, the volumetric heat capacity ρC_p can be assumed to be independent of the temperature (Dulong-Petit law), and the elastic entropy is

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

where α is the thermal expansion coefficient tensor. For an isotropic material, it simplifies into

$$S_{\text{elast}} = \alpha(\sigma_x + \sigma_y + \sigma_z)$$

The heat balance equation can be written as

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

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

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

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

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

$$\rho C_p \frac{\partial T}{\partial t} + \sum_{m,n} T \alpha_{mn} \frac{\partial}{\partial t} \sigma_{mn} = \nabla \cdot (k \nabla T) + Q_h \quad (3-13)$$

In many cases, the second term can be neglected in the left-hand side of [Equation 3-13](#) because all $T \alpha_{mn}$ are small. The resulting approximation is often called uncoupled thermoelasticity.

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 a 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 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 a finite-element sense) to the original pressure calculated from the strains.

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

Select this setting when the material data is close to incompressibility. For an isotropic 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 and the pressure p is calculated from the stress tensor

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

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

$$\frac{p_w}{K} - \frac{p}{K} = 0 \quad (3-14)$$

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 are orders of magnitude smaller.

The modified stress tensor \tilde{s} is then used then in calculations of the energy variation.

When the **Strain formulation** is selected for isotropic linear elastic materials, the auxiliary volumetric strain ϵ_w is used instead of the auxiliary pressure p_w , and it is the set equal to the volumetric strain ϵ_{vol} using the equation

$$-K(\epsilon_w - \epsilon_{\text{vol}}) = 0 \quad (3-15)$$

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 ϵ_w are of the same order of magnitude than 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:

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

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

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

and

$$\boldsymbol{\varepsilon}_{\text{el}} = \text{dev}(\boldsymbol{\varepsilon}_{\text{el}}) + \frac{1}{3}\text{trace}(\boldsymbol{\varepsilon}_{\text{el}})\mathbf{I}$$

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

$$\boldsymbol{\sigma} = \text{dev}(\boldsymbol{\sigma}) + \frac{1}{3}\text{trace}(\boldsymbol{\sigma})\mathbf{I} = 2G\text{dev}(\boldsymbol{\varepsilon}_{\text{el}}) + K\text{trace}(\boldsymbol{\varepsilon}_{\text{el}})\mathbf{I}$$

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

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

The linear relation between pressure and volumetric elastic strain is thus

$$p = -K\boldsymbol{\varepsilon}_{\text{el,vol}}$$

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 $\gamma_{\text{el}} = 2\sqrt{J_2(\boldsymbol{\varepsilon}_{\text{el}})}$ are used alternatively as variables.

In the most general case:

$$p = p(\varepsilon_{\text{el,vol}}) \quad \text{and} \quad \tau = \tau(\gamma_{\text{el}})$$

Tangent and secant moduli

The tangent shear modulus $G_t(\gamma_{\text{el}})$ and the secant shear modulus $G_s(\gamma_{\text{el}})$ in the most general case depend nonlinearly on the shear strain, and are defined as

$$G_t = \frac{\partial \tau}{\partial \gamma_{\text{el}}} \quad \text{and} \quad G_s = \frac{\tau}{\gamma_{\text{el}}}$$

The tangent bulk modulus $K_t(\varepsilon_{\text{el,vol}})$ and the secant bulk modulus $K_s(\varepsilon_{\text{el,vol}})$ depend on the elastic volumetric strain, and are defined as

$$K_t = -\frac{\partial p}{\partial \varepsilon_{\text{el,vol}}} \quad \text{and} \quad K_s = -\frac{p}{\varepsilon_{\text{el,vol}}}$$

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

Ramberg-Osgood material model (Ref. 1) is a nonlinear elastic material commonly used to model plastic deformation in metals, but it also often used in soil engineering. As it is an elastic model, it can only represent plasticity during pure on-loading conditions.

For uniaxial extension, the stress-strain curve is defined by the expression

$$\varepsilon = \frac{\sigma}{E} + \varepsilon_{\text{ref}} \left(\frac{\sigma}{\sigma_{\text{ref}}} \right)^n$$

Here, E means the initial Young's modulus, and ε_{ref} is the strain at a reference stress σ_{ref} . The parameter n is the stress exponent. It is common to use $\varepsilon_{\text{ref}} = 0.002$, so σ_{ref} is the stress at 0.2% strain, typically denoted by the symbol $\sigma_{0.2}$. This parameter has several names depending on the literature: 0.2% offset yield strength, 0.2% proof stress, 0.2% proof strength, or 0.2% yield stress. Typical values for stainless steel are $E = 200$ GPa, $\sigma_{0.2} = 600$ GPa, and $n = 4.8$.

The linear strain is given by

$$\varepsilon_{\text{cl}} = \frac{\sigma}{E}$$

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

$$\varepsilon = \varepsilon_{cl} + \varepsilon_{nl} = \frac{\sigma}{E} + \varepsilon_{ref} \left(\frac{\sigma}{\sigma_{ref}} \right)^n$$

In order to avoid a circular dependence of internal variables, the nonlinear strain ε_{nl} is defined with an auxiliary degree of freedom, so the stress reads $\sigma = E(\varepsilon - \varepsilon_{nl})$.

Ramberg-Osgood material in soil engineering

In soil engineering, it is common to write Ramberg-Osgood material with the stress-strain expression

$$\varepsilon = \frac{\sigma}{E} + \alpha \frac{\sigma_{ref}}{E} \left(\frac{\sigma}{\sigma_{ref}} \right)^n \quad (3-16)$$

so at the reference stress σ_{ref} , the strain is $\varepsilon = (1 + \alpha)\sigma_{ref}/E$. It is common to use $\alpha = 3/7$, so σ_{ref} represents the stress level at which the secant Young's modulus has been reduced to 70% its initial value: $E_{0.7} = E/(1 + \alpha) = 0.7E$. At this reference stress the strain is $\varepsilon = \sigma_{ref}/E_{0.7}$.

POWER LAW

For this type of material the shear stress is related to the shear strain by the strain exponent n and a reference shear strain γ_{ref} (Ref. 2)

$$\tau = G_0 \gamma_{ref} \left(\frac{\gamma}{\gamma_{ref}} \right)^n$$

The secant shear modulus is given by the power law relation

$$G_s = G_0 \left(\frac{\gamma}{\gamma_{ref}} \right)^{n-1}$$

The strain exponent controls the nonlinear deformation:

- For $n > 1$ the material behaves as a dilatant (shear-thickening) solid
- For $n = 1$ the material is linear elastic
- For $0 < n < 1$ the material behaves as pseudoplastic (shear-thinning) solid
- For $n = 0$ the material is perfectly plastic

BILINEAR ELASTIC

The most commonly mentioned model of “bilinear elastic” material is defined with two different bulk moduli for either tension and compression. Commonly, brittle

materials like graphite and ceramics exhibit this behavior. The secant bulk modulus reads:

$$K = K_t \text{ for } \epsilon_{el,vol} > 0$$

and

$$K = K_c \text{ for } \epsilon_{el,vol} < 0$$

where $\epsilon_{el,vol}$ 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 strains (transverse) 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 it corresponds to the third principal strain, $\epsilon_{ax} = \epsilon_3$, when the principal strains are sorted as $\epsilon_1 > \epsilon_2 > \epsilon_3$. The other two strains (transverse) 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 two strain measures can be obtained from the strain tensor or its principal values, the volumetric strain

$$\epsilon_{vol} = \text{trace}(\epsilon) = \epsilon_1 + \epsilon_2 + \epsilon_3$$

and the shear strain,

$$\gamma = \sqrt{2 \text{dev}(\boldsymbol{\varepsilon}) : \text{dev}(\boldsymbol{\varepsilon})} = \sqrt{\frac{2}{3}((\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_1 - \varepsilon_3)^2)}.$$

These are used to define the axial strain variable for multiaxial loading.

The uniaxial test defines then the relation between the axial stress and axial strain as

$$\sigma_{\text{ax}} = E_s \varepsilon_{\text{ax}}$$

here, E_s is the secant Young's modulus, and the axial stress σ_{ax} is considered as a function of axial strain ε_{ax} . Thus

$$E_s = \frac{\sigma_{\text{ax}}}{\varepsilon_{\text{ax}}}$$

At zero strain, the secant Young's modulus is defined as

$$E_s = \left. \frac{d\sigma_{\text{ax}}}{d\varepsilon_{\text{ax}}} \right|_{\varepsilon_{\text{ax}}=0}$$

Assuming a constant Poisson's ratio, the secant shear modulus is then 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 axial strain for multiaxial loading is computed from the shear strain and Poisson's ratio as

$$\varepsilon_{\text{ax}} = \frac{\sqrt{3}\gamma}{2(1+\nu)}$$

When nonsymmetric stress-strain data is available, the axial strain for multiaxial loading is computed from the volumetric strain and Poisson's ratio

$$\varepsilon_{\text{ax}} = \frac{\varepsilon_{\text{vol}}}{1-2\nu}$$

as this expression captures the change of sign in the 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 taken into account. The secant shear modulus is instead defined as

$$G_s = \frac{3KE_s}{9K - E_s}$$

and the axial strain is defined from both shear volumetric and shear strain

$$\varepsilon_{ax} = \frac{\gamma}{\sqrt{3}} + \frac{|\varepsilon_{vol}|}{3}$$



See also the description of the [Uniaxial Data](#) material model in the Solid Mechanics interface documentation.

HYPERBOLIC LAW

An hyperbolic relation between shear stress and shear strain is obtained by setting the secant shear modulus

$$G_s = G \frac{1}{1 + \left(\frac{\gamma}{\gamma_{ref}}\right)^n}$$

where the strain exponent n and a reference shear strain γ_{ref} control the shape of the hyperbola.

For hyperbolic material models, the maximum shear modulus is occurs at zero shear strain, so practitioners might call G the “maximum shear modulus” and use the notation G_{max} . Sometimes it is also called “small strain shear modulus”.

HARDIN-DRNEVICH

The Hardin-Drnevich model ([Ref. 3](#)) is an hyperbolic soil model (with $n = 1$) defined by two input parameters: the initial shear modulus G and a reference shear strain γ_{ref}

$$G_s = G \frac{1}{1 + \frac{\gamma}{\gamma_{ref}}}$$

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 $\tau_{max} = G \gamma_{ref}$ as the shear strain increases.

The hyperbolic Hardin-Drnevich model is normally used for quantifying stiffness reduction curves in soils. Commonly, the reference shear strain γ_{ref} is replaced by the reference shear strain at which the secant shear modulus has been decreased to 70% 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 σ_1 and σ_3 and the axial strain ϵ , and it describes the stress-strain curve by fitting the hyperbola

$$\sigma_1 - \sigma_3 = \frac{\epsilon}{a + b\epsilon}$$

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_{ult}

$$\frac{1}{b} = (\sigma_1 - \sigma_3)_{\text{ult}} = q_{\text{ult}}$$

The ultimate value q_{ult} is related to the strength of the soil.

For the triaxial test, the axial strain ϵ is related to the shear strain γ by the Poisson's ratio as

$$\varepsilon = \frac{\sqrt{3}}{2(1+\nu)}\gamma$$

and the axial and radial stresses are related to the shear stress as $\sigma_1 - \sigma_3 = \sqrt{3}\tau$.

It is possible then to write the relation between shear stress and shear strain as

$$\tau = \frac{\frac{1}{2(1+\nu)}\gamma}{\frac{1}{E} + \frac{1}{q_{ult}} \frac{\sqrt{3}}{2(1+\nu)}\gamma}$$

Since the initial shear modulus is related to the initial Young's modulus as $G = E/2(1+\nu)$, we can alternatively write this stress-strain relation as

$$\tau = \frac{\gamma}{\frac{1}{G} + \frac{\sqrt{3}\gamma}{q_{ult}}}$$

which is an hyperbolic law with a secant shear modulus of

$$G_s = \frac{G}{1 + \frac{G}{q_{ult}}\sqrt{3}\gamma}$$

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 et al. in order to include a nonlinear volumetric response in soils.

The model defines the nonlinear volumetric response for the pressure as

$$p = -K \frac{\varepsilon_{el,vol}}{1 - \frac{\varepsilon_{el,vol}}{\varepsilon_{ult}}}$$

where $\varepsilon_{el,vol}$ is the volumetric strain, and ε_{ult} is the asymptote of the hyperbola, the maximum value for the volumetric compression. Note that K represents the bulk modulus at zero strain.

The secant (nonlinear) bulk modulus is defined for this material model as

$$K_s = K \frac{1}{1 - \frac{\epsilon_{el,vol}}{\epsilon_{ult}}}$$

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 $\epsilon_{el,vol}$ 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.

GENERALIZED MAXWELL MODEL

For isotropic linear elastic materials in the absence of inelastic stresses, Hooke's law in [Equation 3-11](#) reduces to

$$\sigma = \mathbf{C} : \epsilon_{el}$$

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 ϵ_{el} can in the same way be decomposed into volumetric and deviatoric components

$$\epsilon_{el} = \frac{1}{3}\epsilon_{el,vol}I + \epsilon_d$$

with the volumetric elastic strain given by

$$\epsilon_{el,vol} = \text{trace}(\epsilon_{el})$$

and the deviatoric contribution by

$$\epsilon_d = \text{dev}(\epsilon_{el})$$

For isotropic linear elastic materials, the pressure is then related to the volumetric elastic deformation by the bulk modulus K

$$p = -\frac{1}{3}\text{trace}(\sigma) = -K\text{trace}(\epsilon_{el}) = -K\epsilon_{el,vol}$$

and the deviatoric stress tensor is linearly related to the deviatoric elastic strain tensor by the shear modulus G

$$\sigma_d = \text{dev}(\sigma) = 2G\text{dev}(\epsilon_{el}) = 2G\epsilon_d$$

The total stress in [Equation 3-11](#) is then

$$\sigma = K\epsilon_{el,vol}I + 2G\epsilon_d$$

In case of geometric nonlinearity, σ represents the second Piola-Kirchhoff stress tensor and ϵ_{el} the elastic Green-Lagrange strain tensor.

For viscoelastic materials, the deviatoric stress σ_d is not linearly related to the deviatoric strain ϵ_d but it also depends on the strain history. It is normally defined by the hereditary integral:

$$\sigma_d = 2 \int_0^t \Gamma(t-t') \frac{\partial \epsilon_d}{\partial t'} dt'$$

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](#)

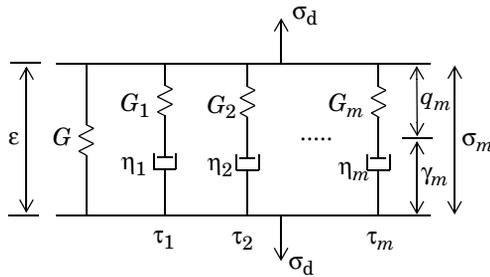


Figure 3-1: Generalized Maxwell model.

Hence, G is the stiffness of the main elastic branch, G_m represents the stiffness of the spring in branch m , and τ_m is the *relaxation time* constant of the spring-dashpot pair in branch m .

The auxiliary strain variable q_m is introduced to represent the extension of the corresponding abstract spring, and the auxiliary variables $\gamma_m = \epsilon - q_m$ 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 is often denoted with the symbol G_∞ . The instantaneous shear modulus G_0 is defined as the sum of the stiffness of all the branches

$$G_0 = G + \sum_{m=1}^{N} G_m$$

This is the stiffness when the external load is applied much faster than the shortest relaxation time of any viscous branch.

The relaxation time τ_m is normally measured in the frequency domain, so the viscosity of the dashpot is not a physical quantity but instead it is derived from stiffness and relaxation time measurements. The viscosity of each branch can be expressed in terms of the shear modulus and relaxation time as

$$\eta_m = G_m \tau_m$$

The stress per branch can be written either in terms of the strain in the spring q_m or the strain in the dashpot γ_m

$$\sigma_m = 2G_m q_m = 2\eta_m \dot{\gamma}_m = 2G_m (\varepsilon - \gamma_m)$$

The sum of the stresses in the viscoelastic branches is then computed from

$$\sigma_q = \sum_{m=1}^{LV} \sigma_m = \sum_{m=1}^{LV} 2G_m (\varepsilon - \gamma_m)$$

The total stress in Hooke's law (Equation 3-11) is then augmented by the viscoelastic stress σ_q

$$\sigma = \sigma_0 + \mathbf{C} : \varepsilon_{el} + \sigma_q \quad (3-17)$$

Computing the Stress on Each Branch

The auxiliary variable γ_m is a symmetric 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 γ_m can be computed by solving the ODE

$$\eta_m \dot{\gamma}_m = G_m (\varepsilon - \gamma_m) \quad (3-18)$$

The relation between viscosity and relaxation time is

$$\eta_m = G_m \tau_m$$

so that Equation 3-18 can equivalently be written as

$$\tau_m \dot{\gamma}_m + \gamma_m = \varepsilon \quad (3-19)$$

The viscoelastic strain variables γ_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. The viscoelastic strain variables do not require continuity so discontinuous shape functions are used.



The viscoelastic strain variables γ_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.

Energy Dissipation

The *dissipated energy density rate* (SI unit: W/m^3) in each branch m is

$$\dot{W}_m = \sigma_m : \dot{\gamma}_m$$

The rate of total dissipated energy density in the Generalized Maxwell material is then

$$\dot{W}_v = \sum_{m=1}^{LV} \dot{W}_m$$

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

$$W_v = \omega G'' \tilde{\epsilon}_d : \text{conj}(\tilde{\epsilon}_d)$$

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:

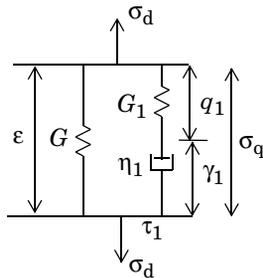


Figure 3-2: Standard linear solid (SLS) model.

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 γ_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

$$\dot{W}_v = \sigma_q : \dot{\gamma}_1$$

KELVIN-VOIGT MODEL

The *Kelvin-Voigt viscoelastic model* is represented by a spring connected in parallel with a damper:

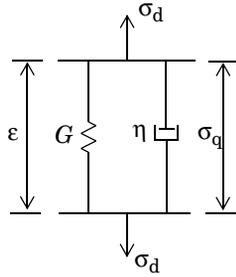


Figure 3-3: Kelvin-Voigt model.

The stress tensor in the viscous branch is computed from the elastic strain rate

$$\sigma_q = 2\eta \dot{\varepsilon} = 2G\tau \dot{\varepsilon} \quad (3-20)$$

so there is no need to add the extra DOFs to compute the auxiliary strain tensor γ .

The relaxation time relates the viscosity and shear modulus by $\eta = \tau G$. The equivalent shear modulus is used in case of a anisotropic linear elastic material.

The dissipated energy density rate of the Kelvin-Voigt model is then computed from its rate

$$\dot{W}_v = \sigma_q : \dot{\varepsilon}$$

BURGERS MODEL

The *Burgers model*, consists of a spring-dashpot branch in series with a Kelvin-Voigt branch:

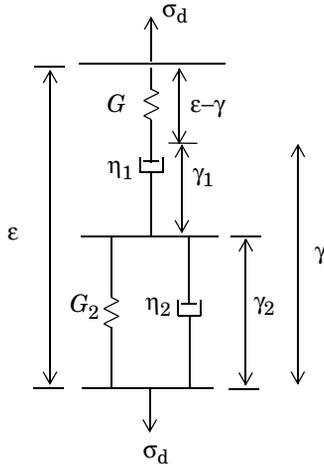


Figure 3-4: Burgers model.

The strain in the first dash-pot follows the ODE

$$2\eta_1\dot{\gamma}_1 = \text{dev}(\sigma) = 2G(\epsilon - \gamma)$$

where the shear modulus G is taken from the parent Linear Elastic material.

The strain in the second dash-pot follows the ODE

$$2\eta_2\dot{\gamma}_2 = \text{dev}(\sigma) - 2G_2\gamma_2$$

The total strain in the dampers is computed from

$$\gamma = \gamma_1 + \gamma_2$$

Combining these equations, it is possible to recover a second-order ODE for the strain tensor γ ,

$$\ddot{\gamma}_2 = \left(\frac{1}{2\eta_1} + \frac{1}{2\eta_2} \right) \text{dev}(\dot{\sigma}) - \frac{G_2}{\eta_2} \dot{\gamma} + \frac{G_2}{2\eta_1\eta_2} \text{dev}(\sigma)$$

Note that Burgers viscoelastic material has two relaxation times related to the stiffness and viscosity in the springs and dash-pots. The relaxation times τ_1 and τ_2 are related to the stiffness and viscosities as $\tau_1 = \eta_1/G$ and $\tau_2 = \eta_2/G_2$.

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 shifts the relaxation time, so Equation 3-19 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 damper.

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 assumed. This shift is only valid over a certain range of temperature, typically around the *glass transition* temperature.

The first step to compute the shift factor α_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 $T - T_0$.



$\alpha_T(T_0) = 1$ so that T_0 is the temperature at which the master curve is given. If the temperature 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 assumed, Q is the activation energy (SI unit: J/mol), and R is the universal gas constant.

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 3-17 and Equation 3-19 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

$$G' = G + \sum_{m=1}^{N} G_m \frac{(\omega\tau_m)^2}{1 + (\omega\tau_m)^2} \quad \text{and} \quad G'' = \sum_{m=1}^{N} G_m \frac{\omega\tau_m}{1 + (\omega\tau_m)^2}$$

and for the SLS model as

$$G' = G + G_1 \frac{(\omega\tau_1)^2}{1 + (\omega\tau_1)^2} \quad \text{and} \quad G'' = G_1 \frac{\omega\tau_1}{1 + (\omega\tau_1)^2}$$

and 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

$$Q_h = \omega G'' \tilde{\varepsilon}_d : \text{conj}(\tilde{\varepsilon}_d)$$



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

$$W_s = W_{\text{iso}} + W_{\text{vol}} + \sum_{m=1}^{IV} \Psi_m$$

The strain energy in the main hyperelastic branch is normally denoted with the superscript ∞ to denote the long-term equilibrium (as $t \rightarrow \infty$).

The second Piola-Kirchhoff stress is computed from

$$S = 2 \frac{\partial W_s}{\partial C} = S_{\text{iso}} + S_{\text{vol}} + \sum_{m=1}^{IV} Q_m$$

where the auxiliary second Piola-Kirchhoff stress tensors Q_m are defined as

$$Q_m = 2 \frac{\partial \Psi_m}{\partial C}$$

The time evolution of the auxiliary stress tensor Q_m in each viscoelastic branch is given by the rate

$$\dot{Q}_m + \frac{1}{\tau_m} Q_m = \dot{S}_{\text{iso},m}$$

here, $S_{\text{iso},m}$ 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* β_m as

$$S_{\text{iso},m} = 2 \frac{\partial W_{\text{iso},m}}{\partial C} = 2 \beta_m \frac{\partial W_{\text{iso}}}{\partial C} = \beta_m S_{\text{iso}}$$

so the time evolution of the auxiliary stress tensor Q_m is given by

$$\dot{Q}_m + \frac{1}{\tau_m} Q_m = \beta_m \dot{S}_{\text{iso}}$$

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

$$\tau_m \dot{q}_m + q_m = \beta_m S_{\text{iso}}$$

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 definition 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 λ_{th} , which for linear thermal expansion in isotropic materials can be written in terms of the isotropic coefficient of thermal expansion, α_{iso} , and the absolute change in temperature

$$J_{th} = \lambda_{th}^3 \text{ and } \lambda_{th} = 1 + \alpha_{iso}(T - T_{ref})$$

Here, the term $\alpha_{iso}(T - T_{ref})$ is the *thermal strain*. The isotropic thermal gradient is therefore a diagonal tensor defined as

$$F_{th} = \lambda_{th}I$$

When the coefficient of thermal expansion α 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 β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. The 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](#)

ISOCORIC 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_{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_{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.Je1`, `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}_{el} in the local coordinate system.
- Principal invariants of the isochoric elastic right Cauchy-Green tensor \overline{C}_{el} .
- Principal invariants of the isochoric elastic Green-Lagrange strain $\overline{\epsilon}_{el}$.

The volumetric strain energy density, W_{vol} can only be defined as an expression of the elastic volumetric deformation J_{el} . The quadratic volumetric strain energy density is defined as:

$$W_{vol}(J_{el}) = \frac{1}{2}\kappa(J_{el} - 1)^2 \quad (3-21)$$

and Harmann-Neff (Ref. 10) volumetric strain energy as:

$$W_{vol}(J_{el}) = \frac{1}{50}\kappa(J_{el}^5 + J_{el}^{-5} - 2) \quad (3-22)$$

where κ is the bulk modulus. From here, the volumetric stress (pressure) is calculated as

$$p_m = -\frac{\partial W_{vol}}{\partial J}$$

When the expression in Equation 3-21 is used, the pressure becomes linearly related to the volume change:

$$p_m = -\kappa(J - 1)$$

but if the expression in Equation 3-22 is used instead of the quadratic function, the pressure becomes a nonlinear function of the volume change:

$$p_m = -\frac{1}{10}\kappa(J^4 - J^{-6})$$

Then, an auxiliary variable, p_w , is added to map the pressure when computing stresses.

The second Piola-Kirchhoff stress is then given by

$$S = -p_w J C^{-1} + 2 \frac{\partial W_{iso}}{\partial C} \quad (3-23)$$

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_{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_{iso} contribution

$$W_s = W_{\text{iso}}$$

The second Piola-Kirchhoff stress is then given by

$$S = -p_w J C^{-1} + 2 \frac{\partial W_{\text{iso}}}{\partial C} \quad (3-24)$$

and the Cauchy stress tensor by

$$\sigma = J^{-1}FSF^T = -p_w I + 2J^{-1}F \frac{\partial W_{\text{iso}}}{\partial C} F^T \quad (3-25)$$

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 3-23](#) 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, λ and μ are the Lamé parameters.

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 κ 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, λ and μ are Lamé parameters. The bulk modulus κ is calculated from the Lamé parameters $\kappa = \lambda + 2\mu/3$.

The nearly incompressible version uses the isochoric invariants $I_1(\overline{\varepsilon}_{el})$ and $I_2(\overline{\varepsilon}_{el})$ to define the isochoric strain energy density

$$W_{iso} = \frac{1}{2}(\lambda + 2\mu)\overline{I}_1^2 - 2\mu\overline{I}_2$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol} , see [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 [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} are related to the Lamé parameter (shear modulus) $\mu = 2(C_{10} + C_{01})$. The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol} , see [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 $I_1(\overline{C}_{el})$ and $I_2(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_{iso} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} C_{mn} (\overline{I}_1 - 3)^m (\overline{I}_2 - 3)^n$$

The elastic volume ratio J_{el} and the bulk modulus κ are used to define the volumetric strain energy density W_{vol} , see [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

$$W_{\text{iso}} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} C_{mn} (\bar{I}_1 - 3)^m (\bar{I}_2 - 3)^n$$

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_{el} and the bulk modulus κ to 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.

YEOH

Yeoh proposed ([Ref. 1](#)) a phenomenological model in order to fit experimental data of filled rubbers, where Mooney-Rivlin and Neo-Hookean models were 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, 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_{el} and the bulk modulus κ 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 [Yeoh](#) material model in the Solid Mechanics interface documentation.

OGDEN

The Neo-Hookean material model usually fits well to experimental data at moderate strains but fails to model hyperelastic deformations at high strains. In order to model rubber-like materials at high strains, Ogden adapted ([Ref. 1](#)) the energy of a Neo-Hookean material to

$$W_s = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p} (\lambda_{el1}^{\alpha_p} + \lambda_{el2}^{\alpha_p} + \lambda_{el3}^{\alpha_p} - 3)$$

Here α_p and μ_p are material parameters, and λ_{el1} , λ_{el2} , and λ_{el3} are the principal elastic stretches such as $J_{el} = \lambda_{el1}\lambda_{el2}\lambda_{el3}$.

The Ogden model is empirical, in the sense that it does not relate the material parameters α_p and μ_p to physical phenomena. The parameters α_p and μ_p are obtained by curve-fitting measured data, which can be difficult for $N > 2$. The most common implementation of Ogden material is with $N = 2$, so four parameters are needed.

The nearly incompressible version uses the isochoric elastic stretches

$$\bar{\lambda}_{eli} = \lambda_{eli} / J_{el}^{1/3}$$

so the isochoric strain energy density is defined as

$$W_{iso} = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p} (\bar{\lambda}_{el1}^{\alpha_p} + \bar{\lambda}_{el2}^{\alpha_p} + \bar{\lambda}_{el3}^{\alpha_p} - 3)$$

The isochoric elastic stretches define a volume preserving deformation, since

$$\bar{\lambda}_{el1}\bar{\lambda}_{el2}\bar{\lambda}_{el3} = \lambda_{el1}\lambda_{el2}\lambda_{el3} / J_{el} = 1$$

The shear modulus is then defined from

$$\mu = \frac{1}{2} \sum_{k=1}^{NV} \alpha_k \mu_k$$

The elastic volume ratio J_{e1} and the bulk modulus κ 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 μ_k 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:

$$W_s = \sum_{k=1}^{NV} \frac{2\mu_k}{\alpha_k^2} \left(\lambda_{cl1}^{\alpha_k} + \lambda_{cl2}^{\alpha_k} + \lambda_{cl3}^{\alpha_k} - 3 + \frac{1}{\beta_k} (J_{e1}^{-\alpha_k \beta_k} - 1) \right)$$

The initial shear and bulk moduli are computed from the parameters μ_k and β_k as

$$\mu = \sum_{k=1}^{NV} \mu_k \quad \text{and} \quad \kappa = \sum_{k=1}^{NV} 2\mu_k \left(\beta_k + \frac{1}{3} \right)$$

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

$$W_s = c_1(\lambda_{cl1} + \lambda_{cl2} + \lambda_{cl3} - 3) + c_2(\lambda_{cl1}\lambda_{cl2} + \lambda_{cl2}\lambda_{cl3} + \lambda_{cl1}\lambda_{cl3} - 3)$$

The nearly incompressible version uses the isochoric elastic stretches defined as

$$\bar{\lambda}_{cli} = \lambda_{cli} / J_{el}^{1/3}$$

so the isochoric strain energy density is defined as

$$W_{iso} = c_1(\bar{\lambda}_{cl1} + \bar{\lambda}_{cl2} + \bar{\lambda}_{cl3} - 3) + c_2(\bar{\lambda}_{cl1}\bar{\lambda}_{cl2} + \bar{\lambda}_{cl2}\bar{\lambda}_{cl3} + \bar{\lambda}_{cl1}\bar{\lambda}_{cl3} - 3)$$

The simplest Varga model is obtained by setting $c_1 = \mu$ and $c_2 = 0$:

$$W_{iso} = \mu(\bar{\lambda}_{cl1} + \bar{\lambda}_{cl2} + \bar{\lambda}_{cl3} - 3)$$

The elastic volume ratio J_{el} and the bulk modulus κ 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, μ_0 is the initial macroscopic shear modulus, $I_1(\mathbf{C}_{el})$ 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 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 ²	19/7000 N ³	519/673750 N ⁴

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, μ_0 and N . This material model is sometimes also called the *eight-chain model* since it was derived for $N = 8$.

The nearly incompressible version uses the isochoric invariant $I_1(\overline{\mathbf{C}}_{el})$ to define the isochoric strain energy density

$$W_{iso} = \mu_0 \sum_{p=1}^5 c_p (\overline{I}_1^p - 3^p)$$

The elastic volume ratio J_{el} and the bulk modulus κ 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, μ and j_m , which defines the strain energy density as:

$$W_s = -\frac{\mu}{2} j_m \log \left(1 - \frac{\overline{I}_1 - 3}{j_m} \right)$$

Here, μ is the shear modulus and j_m is a limiting value for $I_1 - 3$, 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 $I_1(\overline{C}_{el})$ to define the isochoric strain energy density

$$W_{iso} = -\frac{\mu}{2} j_m \log \left(1 - \frac{\overline{I}_1 - 3}{j_m} \right)$$

The elastic volume ratio J_{e1} and the bulk modulus κ 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](#). 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 $I_3(C_{el})$

$$W_s = \phi \frac{\mu}{2} \left((I_1 - 3) + \frac{1}{\beta} (I_3^\beta - 1) \right) + (1 - \phi) \frac{\mu}{2} \left(\left(\frac{I_2}{I_3} - 3 \right) + \frac{1}{\beta} (I_3^\beta - 1) \right)$$

Here, ϕ is an interpolation parameter bounded to $0 < \phi < 1$, the parameter μ is the shear modulus, and β 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 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 are the Murnaghan third-order elastic moduli, which can be found experimentally for many commonly encountered materials such as steel and aluminum, and λ and μ are the Lamé parameters.



See also the description of the [Murnaghan](#) material model in the Solid Mechanics interface documentation.

USER DEFINED

It is possible to define the strain energy density for compressible, nearly incompressible or incompressible hyperelastic materials when selecting the user defined option.

For **Compressible** hyperelastic materials, enter an expression for the elastic strain energy W_s , which can include any expressions involving the following:

- Components of C_{el} , the elastic right Cauchy-Green deformation tensor in the local material coordinate system.
- Principal invariants of C_{el}

$$I_1(C_{el}) = \text{trace}(C_{el})$$

$$I_2(C_{el}) = \frac{1}{2}(I_1^2(C_{el}) - \text{trace}(C_{el}^2))$$

$$I_3(C_{el}) = \det(C_{el})$$



The internal variables for these invariants are named `solid.I1Ce1`, `solid.I2Ce1`, and `solid.I3Ce1`.

- Components of the elastic Green-Lagrange strain tensor ϵ_{el} in the local coordinate system.

- Principal elastic stretches λ_{e11} , λ_{e12} , and λ_{e13} , 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

$$\boldsymbol{\varepsilon}_{el} = \frac{1}{2}(C_{el} - \mathbf{I})$$

the invariants of $\boldsymbol{\varepsilon}_{el}$ are written in terms of the invariants of C_{el} :

$$I_1(\boldsymbol{\varepsilon}_{el}) = \text{trace}(\boldsymbol{\varepsilon}_{el}) = \frac{1}{2}(I_1(C_{el}) - 3)$$

$$I_2(\boldsymbol{\varepsilon}_{el}) = \frac{1}{2}(I_1^2(\boldsymbol{\varepsilon}_{el}) - \text{trace}(\boldsymbol{\varepsilon}_{el}^2)) = \frac{1}{4}(I_2(C_{el}) - 2I_1(C_{el}) + 3)$$

$$I_3(\boldsymbol{\varepsilon}_{el}) = \det(\boldsymbol{\varepsilon}_{el}) = \frac{1}{8}(I_3(C_{el}) - I_2(C_{el}) + I_1(C_{el}) - 1)$$



The internal variables for these invariants are named `solid.I1ee1`, `solid.I2ee1`, and `solid.I3ee1`.

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 $J_{el} = \det(\boldsymbol{F}_{el})$
- The isochoric strain energy, W_{iso} , as an expression involving the invariants of the isochoric elastic right Cauchy-Green tensor $I_1(\overline{C}_{el})$ and $I_2(\overline{C}_{el})$; the invariants of

the isochoric elastic Green-Lagrange strain $I_1(\bar{\boldsymbol{\varepsilon}}_{el})$, $I_2(\bar{\boldsymbol{\varepsilon}}_{el})$, and $I_3(\bar{\boldsymbol{\varepsilon}}_{el})$; or the principal isochoric elastic stretches defined as $\bar{\lambda}_{eli} = \lambda_{eli}/J_{el}^{1/3}$



The internal variables for J_{el} , $I_1(\bar{\boldsymbol{C}}_{el})$, and $I_2(\bar{\boldsymbol{C}}_{el})$ are named `solid.Je1`, `solid.I1C1e1`, and `solid.I2C1e1`.

The internal variables for $I_1(\bar{\boldsymbol{\varepsilon}}_{el})$, $I_2(\bar{\boldsymbol{\varepsilon}}_{el})$, and $I_3(\bar{\boldsymbol{\varepsilon}}_{el})$ 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_{iso} , as done for the **Nearly incompressible material** option. An extra variable is added to enforce the incompressibility condition $J_{el} = 1$, 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 $\boldsymbol{F} = \nabla \mathbf{u} + \mathbf{I}$ tensors, their transpose, inversions, as well as the global material system components of \boldsymbol{C} and $\boldsymbol{\varepsilon}$. 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. 21-23](#)) describes the stress-softening phenomenon observed under cyclic loading in elastomers and biological materials.

Ogden and Roxburgh ([Ref. 24](#)) used an additional state variable to model the Mullins effect. The state variable η is introduced to memorize the microstructural damage on reinforced rubber after repeated loading-unloading cycles. The modified isochoric strain energy density reads

$$\bar{W}_{iso} = \eta W_{iso} + \phi(\eta)$$

here, W_{iso} is the isochoric strain energy of the undamaged material, and $\phi(\eta)$ is referred as the *damage function*. The choice of the damage function $\phi(\eta)$ is completely arbitrary as long as the some constrains are fulfilled. The authors proposed a state function based on the error function, which defined how the state variable η varies as a function of the isochoric strain energy

$$\eta = 1 - \frac{1}{r} \operatorname{erf}\left(\frac{W_{\text{max}} - W_{\text{iso}}}{m}\right)$$

here, $\operatorname{erf}(\cdot)$ is the error function, r and m are positive parameters, and W_{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. 25-26) 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_{∞} , α and W_{sat} are positive parameters.

Miche (Ref. 27) 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_{sat} and d_{∞} are positive parameters.

In this section:

- [Introduction to Small and Large Plastic Strains](#)
- [Plastic Flow for Small Strains](#)
- [Isotropic Plasticity](#)
- [Yield Function](#)
- [Hill Orthotropic Plasticity](#)
- [Isotropic Hardening](#)
- [Plastic Flow for Large Strains](#)
- [Numerical Solution of the Elastoplastic Conditions](#)

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

$$\begin{aligned} \frac{\partial I_1}{\partial \sigma} &= I \\ \frac{\partial J_2}{\partial \sigma} &= \text{dev}(\sigma) \\ \frac{\partial J_3}{\partial \sigma} &= \text{dev}(\sigma) \text{dev}(\sigma) - \frac{2}{3} J_2 I \end{aligned} \tag{3-26}$$

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

$$\dot{\epsilon}_{\text{pvol}} = \text{trace}(\dot{\epsilon}_{\text{p}}) = \lambda \text{trace}\left(\frac{\partial Q_{\text{p}}}{\partial \sigma}\right) = 3\lambda \frac{\partial Q_{\text{p}}}{\partial I_1}$$

A common measure of inelastic deformation is the *effective plastic strain rate*, which is defined from the plastic shear components

$$\dot{\epsilon}_{\text{pe}} = \sqrt{\frac{2}{3} \text{dev}(\dot{\epsilon}_{\text{p}}) : \text{dev}(\dot{\epsilon}_{\text{p}})} \quad (3-27)$$

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 effective 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:

$$F_{\text{y}} = \phi(\sigma) - \sigma_{\text{ys}}$$

where σ_{ys} is the yield stress. The scalar function $\phi(\sigma)$ is called effective stress. The default form of the effective stress is the *von Mises stress*, which is often used in metal plasticity:

$$\sigma_{\text{mises}} = \sqrt{3J_2(\sigma)} = \sqrt{\frac{3}{2} \text{dev}(\sigma) : \text{dev}(\sigma)}$$

Other expressions can be defined, such as *Tresca stress*, *Hill orthotropic plasticity*, or a user defined expression.

The Tresca effective stress is calculated from the difference between the largest and the smallest principal stress

$$\sigma_{\text{tresca}} = \sigma_{\text{p1}} - \sigma_{\text{p3}}$$

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 effective (von Mises) stress σ_{mises} , or other invariants, principal stresses, or stress tensor components.

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)

$$J_2 = \frac{1}{6}((\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2) + \sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2 = k^2$$

or equivalently $\sqrt{J_2} = k$.

The von Mises criterion is implemented as

$$F = \sqrt{3J_2} - \sigma_{ys} = 0$$

where σ_{ys} is the *yield stress level* (yield stress in uniaxial tension).



The effective or von Mises stress ($\sigma_{\text{mises}} = \sqrt{3J_2}$) is available in the variable `solid.mises`, where `solid` is the name of the physics interface node.

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*.

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 effective 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.

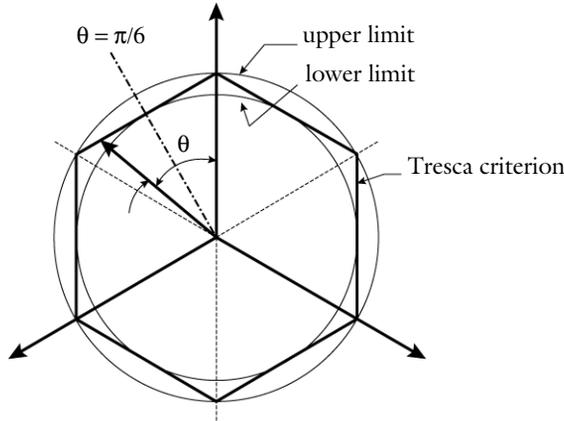


Figure 3-5: The upper and lower limits of the Tresca criterion.

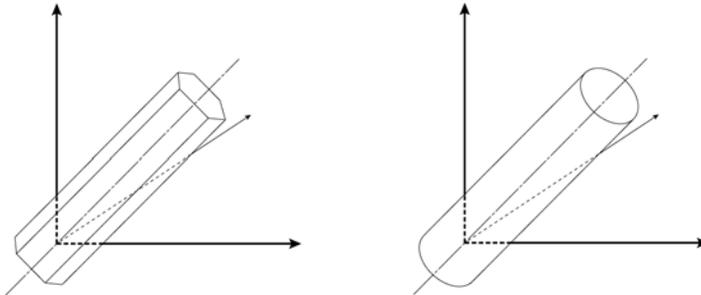


Figure 3-6: Classical yield criteria for metals. Tresca criterion (left) and von Mises criterion (right).

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 belongs to what researchers call *volume preserving* or J_2 plasticity, as the plastic flow is independent on the mean pressure.

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) ϕ 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.

Shima-Oyane Criterion

Shima and Oyane (Ref. 14) 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 effective stress and mean pressure, and it also considers isotropic hardening due to changes in porosity. The plastic potential is defined by

$$Q_p(\sigma) = \left(\frac{\sigma_e}{\sigma_0}\right)^2 + \alpha(1 - \rho_{\text{rel}})^\gamma \left(\frac{p_m}{\sigma_0}\right)^2 - \rho_{\text{rel}}^m$$

here, σ_e is the effective stress, σ_0 is the yield stress, p_m is the pressure, and ρ_{rel} is the relative density. The material parameters α , γ , and m are obtained from curve fitting experimental data. Typical material parameter values for copper aggregates are $\alpha = 6.2$, $\gamma = 1.03$, and $m = 5$.

Gurson Criterion

Gurson criterion (Ref. 15) 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 ϕ , is chosen as main variable. The yield function and associated plastic potential is not an ellipse in the stress space, as in [Shima-Oyane Criterion](#), but it is defined in terms of the hyperbolic cosine function. The plastic potential for Gurson criterion reads

$$Q_p(\sigma) = \left(\frac{\sigma_e}{\sigma_0}\right)^2 + 2\phi \cosh\left(\frac{3p_m}{2\sigma_0}\right) - (1 + \phi^2)$$

here, σ_e is the effective stress, σ_0 is the initial yield stress, p_m is the pressure, and ϕ is the porosity.

Gurson-Tvergaard-Needleman Criterion

Tvergaard and Needleman modified [Gurson Criterion](#) for porous plasticity to include parameters to better fit experimental data ([Ref. 16-17](#)). The resulting criterion is called in the literature Gurson-Tvergaard-Needleman (GTN) criterion. The plastic potential for GTN criterion reads

$$Q_p(\sigma) = \left(\frac{\sigma_e}{\sigma_0}\right)^2 + 2q_1\phi_e \cosh\left(\frac{3q_2p_m}{2\sigma_0}\right) - (1 + q_3\phi_e^2)$$

here, σ_e is the effective stress, σ_0 is the initial yield stress, p_m is the pressure, and ϕ_e is the effective void volume fraction (effective porosity). Typical correction parameter values are $q_1 = 1.5$, $q_2 = 1.03$, and $q_3 = q_1^2$.

The effective void value fraction (or effective porosity) ϕ_e used in the plastic potential is a function of the current porosity ϕ and other material parameters:

$$\phi_e = \begin{cases} \phi & \text{for } \phi < \phi_c \\ \phi_c + \frac{\phi_m - \phi_c}{\phi_f - \phi_c}(\phi - \phi_c) & \text{for } \phi_c < \phi < \phi_f \end{cases}$$

here ϕ_c is the critical void volume fraction (critical porosity) at which void coalescence begins, and ϕ_f is the void volume fraction at failure. When the porosity increases up to the value of failure, the effective porosity takes a maximum value of ϕ_m and the porous material loses the capacity to carry stresses. The maximum porosity value is derived from other parameters

$$\phi_m = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3}$$

Since typical values for the parameters are $q_3 = q_1^2$, the maximum porosity value is given by $\phi_m = 1/q_1$.

Fleck-Kuhn-McMeeking Criterion

The Fleck-Kuhn-McMeeking criterion ([Ref. 19](#)), also called 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 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, σ_e is the effective stress and p_m is the pressure. The *flow strength* of the material under hydrostatic loading, p_f , is computed from

$$p_f = 2.97(1 - \phi) \frac{2\phi_m - \phi}{\phi_m} \sigma_0$$

here, σ_0 is the initial yield stress, and ϕ is the void volume fraction (porosity). The maximum void volume fraction ϕ_m typically takes the value of 36%, the limit of dense random packing of sintered powder.

FKM-GTN Criterion

The FKM-GTN criterion is a combination of the [Fleck-Kuhn-McMeeking Criterion](#) and [Gurson-Tvergaard-Needleman Criterion](#), intended to cover a wider range of porosities (Ref. 20-21). 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.

Void growth

For all 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 ϕ 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.

Other mechanism that can trigger a increase of porosity are nucleation and shear growth. The increment in porosity based on growth nucleation is given by

$$\dot{\phi}_{\text{nucleation}} = \frac{f_N}{s_N \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{\epsilon_{pm} - \epsilon_N}{s_N} \right)^2} \dot{\epsilon}_{pm} \quad \text{for } p < 0$$

here, ϵ_N is the mean strain for nucleation, f_N is the void volume fraction for nucleating particles, and s_N is the standard deviation. Typical values for these parameters are $\epsilon_N = 0.04$, $s_N = 0.1$, 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

$$\dot{\phi}_{\text{shear}} = k_w w \phi \mathbf{n}^D : \dot{\epsilon}_{pl}$$

here, k_w is a material parameter, ϕ is the current porosity, \mathbf{n}^D is a deviatoric tensor coaxial to the stress tensor, and $\dot{\epsilon}_{pl}$ is the plastic strain rate, which depends on the porous plasticity model. The weight w is computed from stress invariants as

$$w = 1 - \frac{27 J_3^2}{4 J_2^3} = \sin^2(3\theta)$$

where θ 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 $\theta = \pi/6$.

SOIL PLASTICITY

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, τ is the shear stress, c the cohesion, and ϕ 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.

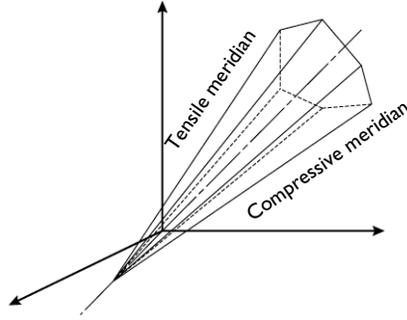


Figure 3-7: The Mohr-Coulomb criterion. The cone opens toward the compressive axis.

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. 9) 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

$$m(\theta) = \sqrt{\frac{1}{3}} \left((1 + \sin \phi) \cos \theta - (1 - \sin \phi) \cos \left(\theta + \frac{2\pi}{3} \right) \right), \quad \alpha = \sin \phi / 3, \quad \text{and}$$

$$k = c \cos \phi$$

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.

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 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)

$$F_y = \sqrt{J_2} + \alpha I_1 - k = 0$$

This is sometimes also called the extended von Mises criterion, since it is equivalent to the von Mises criterion for metals when setting $\alpha = 0$.

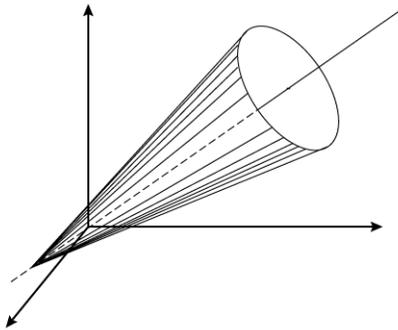


Figure 3-8: The Drucker-Prager criterion. The cone opens toward the compressive axis.

The coefficients in the Drucker-Prager model can be matched to the coefficients in the Mohr-Coulomb criterion by

$$\alpha = \frac{2}{\sqrt{3}} \cdot \frac{\sin \phi}{(3 \pm \sin \phi)} \quad \text{and} \quad k = \frac{2\sqrt{3}c \cos \phi}{(3 \pm \sin \phi)}$$

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

$$m(\pi/3) = (3 - \sin\phi)/(2\sqrt{3})$$

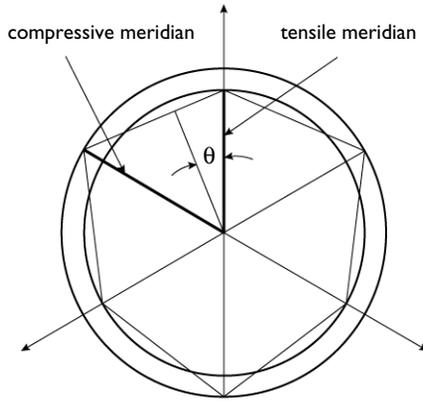


Figure 3-9: The Drucker-Prager criterion showing the tensile and compressive meridians (inner and outer circles), and the Lode angle compared to the cross section of Mohr-Coulomb criterion in the π -plane.

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 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 criteria 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 criteria 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. 5).

Also, when using a Drucker-Prager criterion matched to a Mohr-Coulomb criterion, the plastic potential could also be nonassociated, 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 3-10](#). 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

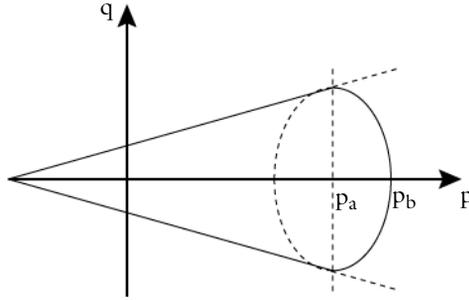


Figure 3-10: Elliptic cap model in Haigh–Westergaard coordinate system.

Note that 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 3-10 shows the cap in terms of the variables p and q

$$q = \sqrt{3J_2(\sigma)} \quad \text{and} \quad p = -I_1(\sigma)/3$$

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

$$p_b = p_{b0} - K_{iso} \log\left(1 + \frac{\varepsilon_{pvol}}{\varepsilon_{pvol,max}}\right)$$

here, p_{b0} is the initial value for the limit pressure p_b , K_{iso} is the isotropic hardening modulus, ϵ_{pvol} the volumetric plastic strain, and $\epsilon_{\text{pvol,max}}$ 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.

Note that the volumetric plastic strain ϵ_{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.

Matsuoka-Nakai Criterion

Matsuoka and Nakai ([Ref. 3](#)) 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 Cauchy 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 ϕ denotes the angle of internal friction in Mohr-Coulomb criterion.

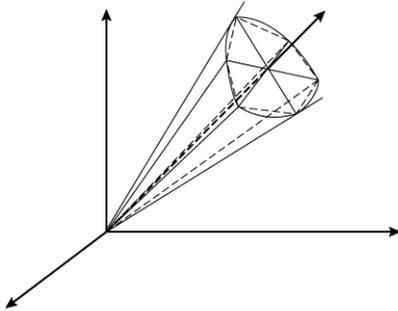


Figure 3-11: The Matsuoka-Nakai criterion and Mohr-Coulomb criterion in the principal stress space.



See also the description of the [Matsuoka-Nakai](#) material model in the Solid Mechanics interface documentation.

Lade-Duncan Criterion

The Lade-Duncan criterion was originally developed to model a large volume of laboratory sample test data of cohesionless soils. 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 ϕ as the angle of internal friction in Mohr-Coulomb criterion

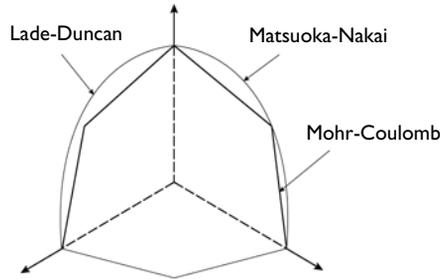


Figure 3-12: Comparing the Mohr-Coulomb, Matsuoka-Nakai, and Lade-Duncan criteria when matching the tensile meridian.



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 Cut-Off

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 the Rankine or *tension cut-off* criterion.

The Rankine criterion states that a material stops deforming elastically when the biggest principal stress σ_1 reaches a maximum tensile stress, also called tension cut-off limit σ_t .

In terms of the principal stress, 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 ϕ . 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 cut-off should be chosen such as

$$\sigma_t < c \frac{\cos \phi}{\sin \phi}$$

The Mohr-Coulomb criterion together with a tension cut-off is sometimes called *modified Mohr-Coulomb criterion* (Ref. 19).



Tension cut-off is also available with Concrete material models.

HILL ORTHOTROPIC PLASTICITY

Hill (Ref. 12, Ref. 13) proposed a quadratic yield function (and associated plastic potential) in a local coordinate system given by the principal axes of orthotropy \mathbf{a}_i

$$Q_p = F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 - 1 \quad (3-28)$$

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

$$\dot{\epsilon}_{\text{pvol}} = \text{trace}(\dot{\epsilon}_{\text{p}}) = \dot{\epsilon}_{\text{p}11} + \dot{\epsilon}_{\text{p}22} + \dot{\epsilon}_{\text{p}33} = 0$$



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

$$L = \frac{1}{2\sigma_{\text{ys}23}^2}, \quad M = \frac{1}{2\sigma_{\text{ys}31}^2}, \quad N = \frac{1}{2\sigma_{\text{ys}12}^2}$$

Here, $\sigma_{\text{ys}ij}$ represents the yield stress in shear on the plane ij .

The material parameters $\sigma_{\text{ys}1}$, $\sigma_{\text{ys}2}$, and $\sigma_{\text{ys}3}$ 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

$$\begin{aligned} \frac{1}{\sigma_{\text{ys}1}^2} &= G + H \\ \frac{1}{\sigma_{\text{ys}2}^2} &= H + F \\ \frac{1}{\sigma_{\text{ys}3}^2} &= F + G \end{aligned}$$

or equivalently

$$\begin{aligned} 2F &= \frac{1}{\sigma_{\text{ys}2}^2} + \frac{1}{\sigma_{\text{ys}3}^2} - \frac{1}{\sigma_{\text{ys}1}^2} \\ 2G &= \frac{1}{\sigma_{\text{ys}3}^2} + \frac{1}{\sigma_{\text{ys}1}^2} - \frac{1}{\sigma_{\text{ys}2}^2} \\ 2H &= \frac{1}{\sigma_{\text{ys}1}^2} + \frac{1}{\sigma_{\text{ys}2}^2} - \frac{1}{\sigma_{\text{ys}3}^2} \end{aligned}$$

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 σ_{ys0} is calculated from the Hill's parameters F , G , and H (this is equivalent to the initial yield stress σ_{ys0} in von Mises plasticity)

$$\frac{1}{\sigma_{ys0}} = \frac{2}{3}(F + G + H) = \frac{1}{3} \left(\frac{1}{\sigma_{ys1}} + \frac{1}{\sigma_{ys2}} + \frac{1}{\sigma_{ys3}} \right) \quad (3-29)$$

Defining *Hill's effective stress* as (Ref. 13)

$$\begin{aligned} \sigma_{hill}^2 = & \sigma_{ys0}^2 (F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 \\ & + 2(L\sigma_{23}^2 + M\sigma_{31}^2 + N\sigma_{12}^2)) \end{aligned}$$

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 σ_{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 σ_{ys0} , the hardening function σ_h , and the effective plastic strain ϵ_{pe} .

ISOTROPIC HARDENING

Plasticity implements seven different kinds of isotropic hardening models for elastoplastic materials:

- [Perfect Plasticity](#) (no isotropic hardening)
- [Linear Isotropic Hardening](#)
- [Ludwik](#)

- Swift
- Voce
- Hockett-Sherby
- User defined

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 effective stress $\phi(\sigma)$ for the yield function from von Mises stress, Tresca stress, Hill effective stress, or a user defined expression; and σ_{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 3-35](#).

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 *effective plastic strain* ϵ_{pe} .

The yield stress $\sigma_{ys}(\epsilon_{pe})$ is then a function of the effective plastic strain and the *initial yield stress* σ_{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 Ludwik model for nonlinear isotropic hardening, the yield stress $\sigma_{ys}(\epsilon_{pe})$ is defined by a nonlinear function of the effective plastic strain. Ludwik equation (also called 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.

Swift

For nonsaturating materials, the Swift power-law equation relates the initial yield stress σ_{ys0} and the isotropic hardening σ_h , to the effective 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 ϵ_0 is a reference strain. Noting that at zero plastic strain the initial yield stress is related to the strength coefficient and hardening exponent as

$$\sigma_{ys}(0) = \sigma_{ys0} = k\epsilon_0^n$$

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 σ_h is exponentially related to the effective 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 β determines the saturation rate of the hysteresis loop for cyclic loading. The *saturation flow stress* σ_{sat} characterizes the maximum distance by which the yield surface can expand in the stress space. For values $\epsilon_{pe} \gg 1/\beta$, the yield stress saturates to

$$\sigma_{ys} \rightarrow \sigma_{ys0} + \sigma_{sat}$$

Hockett-Sherby

The Hockett-Sherby rule for nonlinear isotropic hardening is also intended for materials which yield stress saturates as effective plastic strain increases. It is similar to Voce rule, but it includes an exponential dependency of the form

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + (\sigma_{\infty} - \sigma_{ys0}) \left(1 - e^{-m \epsilon_{pe}^n} \right)$$

where σ_{∞} is the *steady-state flow stress*, m the saturation coefficient and n the saturation exponent. For values $m \epsilon_{pe}^n \gg 1$ the yield stress saturates to

$$\sigma_{ys} \rightarrow \sigma_{\infty}$$

User defined

The yield stress versus the effective plastic strain can be specified with the help of a hardening curve that could also depend on other variables, such as temperature.

In this case, define the (usually nonlinear) hardening function $\sigma_h(\epsilon_{pe})$ such that the yields stress reads

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + \sigma_h(\epsilon_{pe})$$



The internal variable for the effective plastic strain is named `solid.epe`. The effective 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 3-35](#).

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

$$F_y = \phi(\sigma - \sigma_{back}) - \sigma_{ys}$$

Here, σ_{ys} is the yield stress (which may include a linear or nonlinear isotropic hardening model), and the effective 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*, σ_{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 σ_{back} is a linear function of the plastic strain tensor ϵ_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. 7) added memory to Prager's linear kinematic hardening model. This nonlinear kinematic hardening model allows to capture the Bauschinger effect and nonlinear behavior by non symmetrical tension-compression loading.

The nonlinear evolution of the back stress σ_{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, γ_k is a kinematic hardening parameter, and ϵ_{pe} the effective 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 ϵ_{back} as proposed in (Ref. 8), 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. 8) 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

$$\dot{\sigma}_{\text{back},i} = \frac{2}{3}C_i \dot{\epsilon}_p - \gamma_i \dot{\epsilon}_{pe} \sigma_{\text{back},i}$$

Practitioners would normally select $\gamma_k = 0$ for one of the back stress equations, thus recovering Prager's linear rule for that branch

$$\sigma_{\text{back},0} = \frac{2}{3}C_0 \epsilon_p$$

The back stress tensor σ_{back} is then defined by the superposition of N back stress tensors

$$\sigma_{\text{back}} = \frac{2}{3}C_0 \epsilon_p + \sum_{i=1}^N \sigma_{\text{back},i}$$

As done for Armstrong-Frederik kinematic hardening, the algorithm solves for the back strain tensors $\epsilon_{\text{back},i}$ instead of the back stress tensors. The change of variables is

$$\sigma_{\text{back},i} = \frac{2}{3}C_i \epsilon_{\text{back},i}$$

and the nonlinear evolution for the back strain tensors reads

$$\dot{\epsilon}_{\text{back},i} = \dot{\epsilon}_p - \gamma_i \dot{\epsilon}_p \epsilon_{\text{back},i}$$

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, σ , 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, λ 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 λ is determined by the *complementarity* or *Kuhn-Tucker conditions*

$$\lambda \geq 0, F_y \leq 0 \text{ and } \lambda F_y = 0 \quad (3-30)$$

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 3-31](#) is solved together with the conditions in [Equation 3-30](#).

$$\dot{\epsilon}_p = \lambda \frac{\partial F_y}{\partial \sigma} \quad (3-31)$$

For a *nonassociated* flow rule, the yield function does not coincide with the plastic potential, and together with the conditions in [Equation 3-30](#), the rate in [Equation 3-32](#) is solved for the plastic potential Q_p (often, a smoothed version of F_y).

$$\dot{\epsilon}_p = \lambda \frac{\partial Q_p}{\partial \sigma} \quad (3-32)$$

The evolution of the plastic strain tensor $\dot{\epsilon}_p$ (with either [Equation 3-31](#) or [Equation 3-32](#), plus the conditions in [Equation 3-30](#)) is implemented at Gauss points in the plastic element `elplastic`.

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. 9, Ref. 10, and Ref. 11) 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} :

$$-\frac{1}{2}\mathcal{L}(B_{el}) = \lambda \frac{\partial \Phi}{\partial \tau} B_{el} \quad (3-33)$$

The plastic multiplier λ and the yield function Φ (written in terms of the Kirchhoff stress tensor τ) 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 Φ in Ref. 9 and Ref. 10 was written in terms of Kirchhoff stress τ and not Cauchy stress σ because the authors defined the plastic dissipation with the conjugate energy pair τ 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

$$\mathcal{L}(B_{el}) = FC_p^{-1} \dot{F}^T \quad (3-34)$$

By using Equation 3-33 and Equation 3-34, the either associated or nonassociated plastic flow rule for large strains is written as (Ref. 10)

$$-\frac{1}{2}FC_p^{-1} \dot{F}^T = \lambda \frac{\partial Q_p}{\partial \tau} B_{el} \quad (3-35)$$

together with the Kuhn-Tucker conditions for the plastic multiplier λ and the yield function F_y

$$\lambda \geq 0, F_y \leq 0 \text{ and } \lambda F_y = 0 \quad (3-36)$$

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

$$\dot{C}_p^{-1} = -2\lambda F^{-1} \frac{\partial Q_p}{\partial \tau} F C_p^{-1} \quad (3-37)$$

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 3-37 are replaced by

$$C_p^{-1} = F_p^{-1} F_p^{-T}, \quad \dot{C}_p^{-1} = \dot{F}_p^{-1} F_p^{-T} + F_p^{-1} \dot{F}_p^{-T} \quad \text{and} \quad B_{el} = F F_p^{-1} F_p^{-T} F^T$$

After integrating the flow rule in Equation 3-37, 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 effective plastic strain variable is computed as the *true* effective 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 Δt is the pseudo-time step length.

For large plastic strains, Equation 3-37 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 (ε_p or F_p^{-1} , depending on whether small strain or large strain plasticity is selected) and the plastic multiplier, Λ , 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. 9):

- 1 Elastic-predictor: Try the elastic solution $\varepsilon_p = \varepsilon_{p, \text{old}}$ (or $F_p^{-1} = F_{p, \text{old}}^{-1}$) 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 Material

In this section:

- [Bresler-Pister Criterion](#)
- [Willam-Warnke Criterion](#)
- [Ottosen Criterion](#)
- [Original Hoek-Brown Criterion](#)
- [Generalized Hoek-Brown Criterion](#)

BRESLER-PISTER CRITERION

The Bresler-Pister criterion (Ref. 2, Ref. 17) 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

$$F_y = \sqrt{J_2} + k_1 I_1^2 + k_2 I_1 + k_3 \quad (3-38)$$

here, k_1 , k_2 , and k_3 are parameters obtained from the uniaxial compressive strength σ_c , the uniaxial tensile strength σ_b , and the biaxial compressive strength σ_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.

WILLAM-WARNKE CRITERION

The Willam-Warnke criterion (Ref. 10) 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

$$F_y = \sqrt{\frac{3}{5}} \frac{\tau_{\text{oct}}}{\sigma_c} + r(\theta) \left(\left(\frac{1}{\sigma_t} - \frac{1}{\sigma_b} \right) \sigma_{\text{oct}} - 1 \right) \quad (3-39)$$

where σ_c is the uniaxial compressive strength, σ_t is uniaxial tensile strength, and σ_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 3-39](#) 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 θ

$$r(\theta) = \frac{2r_c(r_c^2 - r_t^2) \cos \theta + r_c(2r_t - r_c) \sqrt{4(r_c^2 - r_t^2) \cos^2 \theta + 5r_t^2 - 4r_c r_t}}{4(r_c^2 - r_t^2) \cos^2 \theta + (r_c - 2r_t)^2}$$

Here, the tensile and compressive meridian r_t and r_c are defined in terms of the strengths σ_c , σ_b , and σ_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 θ (Ref. 10).

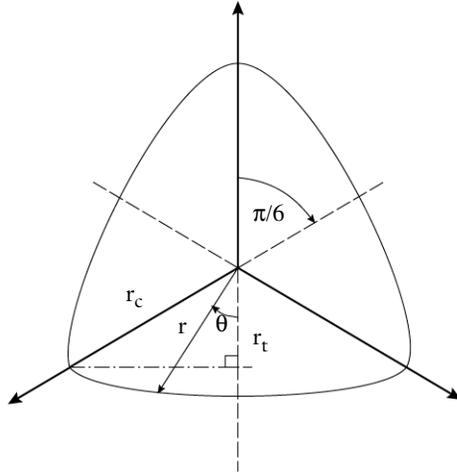


Figure 3-13: The deviatoric section of Willam-Warnke failure criterion.

OTTOSEN CRITERION

The Ottosen criterion is a four-parameter failure criterion proposed for short-time loading of concrete. It corresponds to a smooth convex failure surface with curved meridians, which is open in the negative (compressive) direction of the hydrostatic axis. The 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. 18).

The Ottosen criterion is commonly written as (Ref. 17, Ref. 18):

$$F_y = \frac{a}{\sigma_c} J_2 + \lambda(\theta) \sqrt{J_2} + b I_1 - \sigma_c$$

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 θ 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. 17, Ref. 18).

Typical values for these parameters are obtained by curve-fitting the uniaxial compressive strength σ_c , uniaxial tensile strength σ_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 σ_c , σ_b , and σ_t are positive.

TABLE 3-3: TYPICAL PARAMETER VALUES FOR OTTOSEN FAILURE CRITERION (Ref. 18).

σ_t / σ_c	a	b	k_1	k_2	λ_t	λ_c
0.08	1.808	4.096	14.486	0.991	14.472	7.783
0.10	1.276	3.196	11.736	0.980	11.711	6.531
0.12	0.922	2.597	9.911	0.965	9.872	5.698

The compressive and tensile meridians (as defined in the [Willam-Warnke Criterion](#)) are

$$r_c = \frac{1}{\lambda_c} = \frac{1}{\lambda(\pi/3)} \quad \text{and} \quad r_t = \frac{1}{\lambda_t} = \frac{1}{\lambda(0)}$$

The ratio $r_t/r_c = \lambda_c/\lambda_t$ normally lies between 0.54-0.58 for concrete.



The Ottosen criterion is equivalent to Drucker-Prager when $a = 0$ and λ independent of the Lode angle.

ORIGINAL HOEK-BROWN CRITERION

The Hoek-Brown criterion is an empirical type of model which is commonly used when dealing with rock masses of variable quality. The Hoek-Brown criterion is widely used within civil engineering and is popular because the material parameters can be estimated based on simple field observations together with knowledge of the uniaxial compressive strength of the intact rock material. The Hoek-Brown criterion is one of the few nonlinear criteria widely accepted and used by engineers to estimate the yield and failure of rock masses. The original Hoek-Brown failure criterion states (Ref. 5)

$$\sigma_1 = \sigma_3 + \sqrt{m\sigma_c\sigma_3 + s\sigma_c^2}$$

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*), σ_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 σ_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. 7).

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

$$F_y = 2\sqrt{J_2} \sin\left(\theta + \frac{\pi}{3}\right) - \sigma_c \sqrt{s - m \frac{\sigma_1}{\sigma_c}} = 0$$

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. 6). 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 :

$$m_b = m_i \exp\left(\frac{GSI - 100}{28 - 14D}\right)$$

s and a are positive parameters for the rock mass given by the following relationships:

$$s = \exp\left(\frac{GSI - 100}{9 - 3D}\right)$$

$$a = \frac{1}{2} + \frac{1}{6} \left(\exp\left(\frac{-GSI}{15}\right) - \exp\left(\frac{-20}{3}\right) \right)$$

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

Elastoplastic Soil Models

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 different modifications. The *modified Cam-Clay model* 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 follows different trajectories in stress space. The model also features hardening and softening of clays. Different formulations can be found in textbooks about the model (see [Ref. 13](#), [Ref. 14](#), and [Ref. 15](#)).

The yield function is written in terms of the stress invariants

$$q = \sqrt{3J_2(\boldsymbol{\sigma})} \text{ and } p = -I_1(\boldsymbol{\sigma})/3$$

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 p - q 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 p - q space. This parameter can be related to the angle of internal friction ϕ in the Mohr-Coulomb criterion as

$$M = \frac{6 \sin(\phi)}{3 - \sin(\phi)}$$

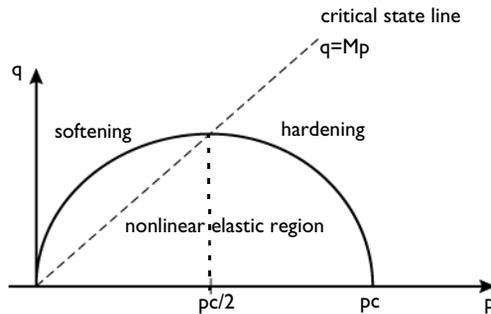


Figure 3-14: Modified Cam-Clay ellipse in the pq -plane. The ellipse circumscribes a nonlinear elastic region.

In the modified Cam-Clay model, hardening is controlled by the *consolidation pressure* p_c , which depends on the volumetric plastic strain $\epsilon_{p,\text{vol}}$ as

$$\dot{p}_c = -\frac{(1 + e_{\text{ref}})}{\lambda - \kappa} p_c \dot{\epsilon}_{p,\text{vol}} \quad (3-40)$$



The volumetric plastic strain is available in the variable `solid.epvol` and the consolidation pressure in the variable `item.pc`.

The evolution of the consolidation pressure depends on the values for the *reference void ratio* e_{ref} , the *swelling index* κ , and the *compression index* λ , which are positive parameters that fulfill

$$0 < \kappa < \lambda \quad \text{and} \quad e_{\text{ref}} > 0$$



The void ratio e is the ratio between pore volume and solid volume. It can be written in terms of the porosity ϕ as $e = \phi / (1 - \phi)$.

The compression index λ is the slope of the *virgin isotropic consolidation line*, and the swelling index κ is the slope of the *rebound-reloading line* (also called loading-reloading line) in the e versus $\ln(p)$ plane.

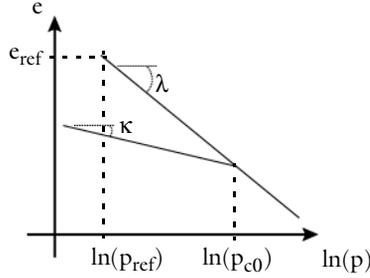


Figure 3-15: Slopes of the virgin isotropic consolidation line and rebound-reloading line in the e vs. $\ln(p)$ plane. The reference void ratio e_{ref} is measured at the reference pressure p_{ref} .



If an Initial Stress and Strain feature is added to the Cam-Clay material, the initial consolidation pressure p_{c0} must be made equal or bigger 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

$$\boldsymbol{\sigma} - \boldsymbol{\sigma}_0 = \mathbf{C}:(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\text{inel}})$$

Here, $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\varepsilon}$ is the total strain tensor, $\boldsymbol{\varepsilon}_{\text{inel}}$ is the inelastic strain tensor, $\boldsymbol{\sigma}_0$ is the initial or external stress tensor, and \mathbf{C} is the fourth-order elasticity tensor. Assuming only elastic stresses in a linear isotropic elastic medium, Hooke's law simplifies to

$$\boldsymbol{\sigma} = K \text{trace}(\boldsymbol{\varepsilon}_{\text{el}}) \mathbf{I} + 2G \text{dev}(\boldsymbol{\varepsilon}_{\text{el}})$$

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, the trace of the stress tensor is linearly related to the volumetric elastic strain $\varepsilon_{\text{el,vol}}$ (the trace of the elastic strain tensor) by the bulk modulus

$$p = -I_1(\boldsymbol{\sigma})/3 = -K\varepsilon_{\text{el,vol}}$$

here, the pressure p is positive in compression, but the elastic volumetric strain $\varepsilon_{\text{el,vol}}$ is positive in tension.



The elastic volumetric strain is available in the variable `solid.eelvol`.

As opposed to a [Linear Elastic Material](#), the modified Cam-Clay soil model introduces a nonlinear relation for the hydrostatic pressure as a function of the volumetric elastic strain:

$$p_{\text{mcc}} = p_{\text{ref}} e^{-B_{\text{el}}\varepsilon_{\text{el,vol}}} \quad \text{with } B_{\text{el}} = \frac{1 + e_{\text{ref}}}{\kappa}$$

here, the reference pressure p_{ref} is the pressure at which the reference void ratio e_{ref} was measured, and κ is the *swelling index*.

The initial bulk modulus is then given by the expression $K_0 = B_{\text{el}}p_{\text{ref}}$, and the tangent bulk modulus by the expression $K_t = B_{\text{el}}p_{\text{mcc}}$. See also the section [Tangent and secant moduli](#).

In order to have zero stress at zero volumetric strain, the reference pressure is removed from the stress tensor, so the total pressure finally reads

$$p = p_{\text{mcc}} - p_{\text{ref}}$$

The stress tensor is then computed from

$$\boldsymbol{\sigma} = (p_{\text{ref}} - p_{\text{mcc}})\mathbf{I} + 2G\text{dev}(\boldsymbol{\varepsilon}_{\text{el}})$$

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 ($\mathbf{Q}_p = \mathbf{F}_y$) give the rate equation for the plastic strains. The plastic strain tensor $\boldsymbol{\varepsilon}_p$ is calculated from the plastic multiplier λ_p and the derivatives of the plastic potential Q_p with respect to the stress tensor $\boldsymbol{\sigma}$

$$\dot{\epsilon}_p = \lambda_p \frac{\partial Q_p}{\partial \sigma} = \lambda_p \left(\frac{\partial Q_p}{\partial I_1} \frac{\partial I_1}{\partial \sigma} + \frac{\partial Q_p}{\partial J_2} \frac{\partial J_2}{\partial \sigma} \right)$$



Here, λ_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

$$\partial I_1 / \partial \sigma = \mathbf{I} \quad \text{and} \quad \partial J_2 / \partial \sigma = \text{dev}(\sigma)$$

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* $\dot{\epsilon}_{p\text{vol}}$) then reads

$$\dot{\epsilon}_{p\text{vol}} = \text{trace}(\dot{\epsilon}_p) = \lambda_p M^2 (p_c - 2p)$$

This relation explains the reason why there is *isotropic hardening* when the pressure is $p > p_c/2$ and *isotropic softening* when $p < p_c/2$. As opposed to what happens in J_2 plasticity, in the modified Cam-Clay soil model the volumetric plastic strain can either increase or decrease as plastic deformation occurs.

In the modified Cam-Clay model, hardening is controlled by the consolidation pressure p_c as a function of volumetric plastic strain, as described in [Equation 3-40](#).

Hardening introduces changes in the shape of the Cam-Clay ellipse, since its major semi-axis depends on the value of the consolidation pressure p_c . The *initial consolidation pressure* p_{c0} defines the size of the ellipse before plastic deformation occurs.

Including pore pressure

When an external pore pressure p_f is added to the modified Cam-Clay material, the yield function is shifted on the p axis, and the yield function reads:

$$F_y = q^2 + M^2(p - p_f - p_c)(p - p_f)$$

The quantity $p - p_f$ is normally regarded as the *effective pressure*, or effective stress, which should not be confused with von Mises stress. To add the effect of a fluid pressure in the pores p_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. 20](#), [Ref. 21](#)) 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. 20](#)), 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 $\epsilon_{p,dev}$. 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

$$\begin{aligned} p_b &= p_{bi} e^{-\epsilon_{p,dev}} && \text{if } \epsilon_{p,dev} \leq \epsilon_{p,devf} \\ p_b &= p_{bf} e^{-d_s(\epsilon_{p,dev} - \epsilon_{p,devf})} && \text{otherwise} \end{aligned} \quad (3-41)$$

here, p_{bi} is the initial structural strength, p_{bf} the failure structural strength, $\epsilon_{p,devc}$ is the effective plastic strain at failure, and d_s is the destructuring index due to shear deformation.



The effective deviatoric plastic strain at which the crushing of the structure begins, $\epsilon_{p,devc}$, has a typical value between 0.15 and 0.3 for most clays.

Structured clays show a higher void ratio than destructured clays at the same effective mean stress. The virgin compression behavior during the destructuring process is expressed by

$$e = e^* + \Delta e_i (p_{c0}/p_c)^{d_v}$$

where e is void ratio of the structured clay, e^* is void ratio of the destructured clay at the same stress state, Δe_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

$$F_y = q^2 + M(\theta)^2(p + p_b)(p - p_c)$$

The nonassociated plastic potential reads

$$Q_p = q^2 + \frac{M(\theta)^2}{\zeta - 1} \left((p + p_b)^2 - \left(\frac{p + p_b}{p_c + p_b} \right)^{2/\zeta} (p_c + p_b)^2 \right)$$

here, ζ is a parameter to smooth the shape of the plastic potential.

As opposed to [The Modified Cam-Clay Soil Model](#), the slope of the critical state line, M , depends on both the Lode angle θ and the angle of internal friction ϕ

$$M(\theta) = \left(\frac{6 \sin(\phi)}{3 - \sin(\phi)} \right) \left(\frac{2\omega}{1 + \omega - \sin(3\theta)(1 - \omega)} \right)^{1/4} \quad \text{where } \omega = \left(\frac{3 - \sin(\phi)}{3 + \sin(\phi)} \right)^4$$

In the modified structured Cam-Clay soil model, hardening is controlled by the consolidation pressure p_c as a function of volumetric plastic strain, as described in

Equation 3-40 and the contribution to the mean stress due to the structure, p_b , as written in Equation 3-41.



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](#)

$$p_{\text{mscc}} = p_{\text{ref}} e^{-B_{\text{el}} \varepsilon_{\text{el,vol}}} \quad \text{with } B_{\text{el}} = \frac{1 + e_{\text{ref}}}{\kappa}$$

In order to have zero stress at zero volumetric strain, the reference pressure is removed from the stress tensor, so the total pressure finally reads

$$p = p_{\text{mscc}} - p_{\text{ref}}$$

The stress tensor is then computed from

$$\sigma = (p_{\text{ref}} - p_{\text{mscc}}) \mathbf{I} + 2G \text{dev}(\varepsilon_{\text{el}})$$

THE EXTENDED BARCELONA BASIC SOIL MODEL

The Barcelona Basic model (BBM) was developed to simulate unsaturated and partially saturated soils, by incorporating an extra state variable for the pore suction. Suction affects the flow in porous soils, as well as the stress distribution and deformation.

The suction value depends on the amount of water in the soil. The BBM uses the concepts of plasticity theory, incorporating the critical state model ([Ref. 22](#)). 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 the numerical limitations of the original BBM model ([Ref. 23](#)). The BBMx model presents a smooth yield surface with respect to both stress and suction, and it is the one implemented in the Geomechanics Module.

The yield surface for the extended Barcelona Basic Model reads

$$F_y = q^2 + M(\theta)^2(p - p_{cs})(p + p_s) + p_{\text{ref}}^2 \left(e^{\frac{b(s-s_y)}{p_{\text{ref}}}} - e^{\frac{-bs_y}{p_{\text{ref}}}} \right)$$

here, p_{cs} is the consolidation pressure at current suction, b is a dimensionless smoothing parameter, and s_y is the yield value at current suction. 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 associated plastic potential for the BBMx model reads

$$Q_p = F_y$$

but the plastic strain increments are computed from the derivatives of the plastic potential with respect to stress only.

As described in the modified Cam-Clay model, hardening is controlled by the *consolidation pressure* p_c , which depends on the volumetric plastic strain $\epsilon_{p,\text{vol}}$ as

$$\dot{p}_c = -\frac{(1 + e_{\text{ref}})}{\lambda - \kappa} p_c \dot{\epsilon}_{p,\text{vol}} \quad (3-42)$$

The evolution of the consolidation pressure depends on the values for the *reference void ratio* e_{ref} , the *swelling index* κ , and the *compression index* λ , which 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

$$\dot{s}_y = -\frac{(1 + e_{\text{ref}})}{\lambda_s - \kappa} (s_y + p_{\text{atm}}) \dot{\epsilon}_{p,\text{vol}} \quad (3-43)$$

where λ_s is the compression index at current suction and p_{atm} is the atmospheric pressure. Note that Equation 3-42 and Equation 3-43 are normally given with opposite sign, but here we use Structural Mechanics convention so the increments in consolidation pressure and suction are positive in compression.

The consolidation pressure at current suction p_{cs} is given by

$$p_{cs} = p_{\text{ref}} \left(\frac{p_c}{p_{\text{ref}}} \right)^{\left(\frac{\lambda_0 - \kappa}{\lambda_s - \kappa} \right)}$$

where λ_0 is the compression index at saturation and the compression index at current suction, λ_s , is given by

$$\lambda_s = \lambda_0((1-w)e^{-s/m} + w)$$

where w and m are weighting and soil stiffness parameters.

In the BBMx model, the suction s is linearly related to the volumetric elastic strain $\epsilon_{el,vol}$ and the pressure

$$s = s_0 + K_c \left(\epsilon_{el,vol} - \frac{p}{K} \right)$$

here K is the bulk modulus, K_c the stiffness to suction, and s_0 is the initial value of suction. The pressure p is positive in compression, but the elastic volumetric strain $\epsilon_{el,vol}$ is positive in tension.

The relationship between hydrostatic pressure and volumetric elastic strain is the same as given in the [The Modified Cam-Clay Soil Model](#)

$$p_{bbm} = p_{ref} e^{-B_{el} \epsilon_{el,vol}} \quad \text{with } B_{el} = \frac{1 + e_{ref}}{\kappa}.$$

In order to have zero stress at zero volumetric strain, the reference pressure is removed from the mean stress, so the total pressure finally reads

$$p = p_{bbm} - p_{ref}$$

The stress tensor is then computed from

$$\sigma = (p_{ref} - p_{bbm})I + 2G \text{dev}(\epsilon_{cl})$$

THE HARDENING SOIL MODEL

The Hardening Soil model is a nonlinear elastic material model with stress dependent and stress path dependent stiffness approach. It is a so-called double stiffness model, which has different stiffness during primary loading and unloading/reloading cases ([Ref. 24](#)). The yield surface for the Hardening Soil model is combination of a conical surface and a elliptic cap surface in stress space. The failure in shear occurs according to Mohr-Coulomb criterion.

The stiffness modulus for primary loading, denoted by E_{50} , and for unloading/reloading, denoted by E_{ur} , are given by the expressions

$$E_{50} = E_{50}^{\text{ref}} \left(\frac{c \cot(\phi) - \sigma_1}{c \cot(\phi) + p_{\text{ref}}} \right)^m$$

$$E_{\text{ur}} = E_{\text{ur}}^{\text{ref}} \left(\frac{c \cot(\phi) - \sigma_1}{c \cot(\phi) + p_{\text{ref}}} \right)^m$$

here E_{50}^{ref} and $E_{\text{ur}}^{\text{ref}}$ are reference stiffness moduli at reference pressure p_{ref} , c is the cohesion, ϕ is the angle of internal friction, and m is the stress exponent. From the Mohr-Coulomb criterion, the ultimate deviatoric stress, q_f , and the stress to failure q_a are defined as

$$q_f = \frac{k - \alpha I_1}{m_c} \quad \text{and} \quad q_a = \frac{q_f}{R_f}$$

where R_f is failure ratio.

The yield function and plastic potential for the shear hardening cone are written in terms of stress invariants as

$$F_y = q_a \left(\frac{q_a \sqrt{3J_2}}{E_{50}(q_a - \sqrt{3J_2})} - \frac{2\sqrt{3J_2}}{E_{\text{ur}}} - \gamma_p \right)$$

$$Q_p = \sqrt{\frac{3J_2}{4}} + \left(\frac{I_1}{3} - \sqrt{\frac{J_2}{12}} \right) \sin(\psi_m)$$

here, γ_p is the accumulated plastic shear strain, and ψ_m is the mobilized dilatancy angle.

The yield function, and associated plastic potential, for the elliptic cap is written in terms of stress invariants as

$$F_y = Q_p = \frac{3R_c^2 J_2}{p_c} + \frac{p^2}{p_c} - p_c$$

here, R_c is ellipse aspect ratio.

The evolution of the internal variables p_c and γ_p are governed by the volumetric plastic strain $\epsilon_{\text{pl,vol}}$ and the plastic strain invariant $J_2(\epsilon_{\text{pl}})$ as

$$\dot{p}_c = -H \dot{\epsilon}_{\text{pl,vol}}$$

$$\dot{\gamma}_p = 2\sqrt{J_2}(\epsilon_{\text{pl}})$$

here, H is the hardening modulus, which is derived from the bulk modulus in compression (for primary loading) K_c , and the bulk modulus in swelling (for unloading and reloading) K_s .

$$\frac{1}{H} = \frac{1}{K_c} - \frac{1}{K_s} \quad \text{where } K_s = \frac{E_{ur}^{ref}}{3(1-2\nu_{ur})}$$

The dilatancy cut-off is implemented by setting the mobilized dilatancy angle ψ_m equal to zero when the void ratio reaches the critical void ratio e_{max} .

For additive decomposition of strains, Cauchy's stress tensor is written as

$$\boldsymbol{\sigma} - \boldsymbol{\sigma}_0 = \mathbf{C}:(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{inel})$$

where \mathbf{C} is a function of the stiffness modulus E_{ur} and Poisson's ratio ν .

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:

$$\dot{\boldsymbol{\varepsilon}}_{cr} = \mathbf{F}_{cr}(\boldsymbol{\sigma}, T, t)$$

It is often possible to separate these effects as shown in this equation:

$$\mathbf{F}_{cr}(\boldsymbol{\sigma}, t, T) = f_1(\boldsymbol{\sigma})f_2(T)f_3(t)$$

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.

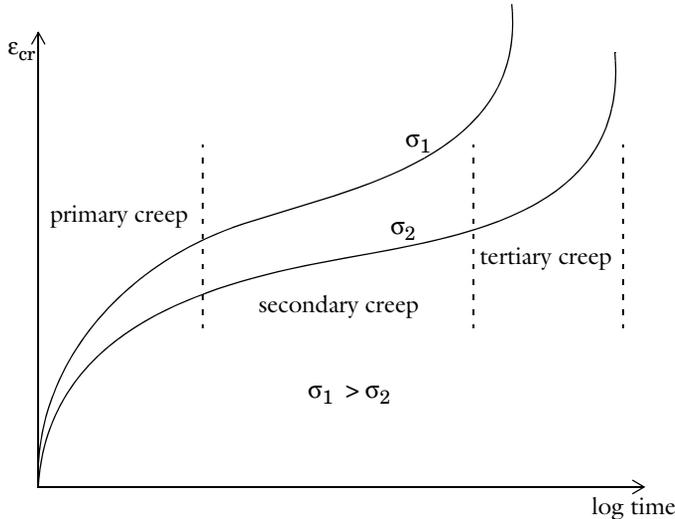


Figure 3-16: Uniaxial creep as a function of logarithmic time.

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)
- 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 effective stress $\sigma_e = \sqrt{3J_2}$.

When (in SI units) the creep potential, Q_{cr} , is given in units of Pa, the rate multiplier η 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

$$\dot{\boldsymbol{\epsilon}}_{\text{cr}} = \dot{\epsilon}_{\text{cre}} \mathbf{n}^D$$

Here, \mathbf{n}^D is a deviatoric tensor coaxial to the stress tensor.

The *effective creep strain rate*, $\dot{\epsilon}_{\text{cre}}$, normally depends on the second deviatoric invariant of the stress $J_2(\boldsymbol{\sigma})$ or the effective or von Mises (effective) stress σ_e , in addition to the temperature and other material parameters.

The deviatoric tensor \mathbf{n}^D is defined as

$$\mathbf{n}^D = \frac{3 \text{dev}(\boldsymbol{\sigma})}{2 \sigma_e} \quad (3-44)$$

The resulting creep strain rate tensor is also deviatoric, since $\text{trace}(\mathbf{n}^D) = 0$

$$\text{trace}(\dot{\boldsymbol{\epsilon}}_{\text{cr}}) = \dot{\epsilon}_{\text{cre}} \text{trace}(\mathbf{n}^D) = 0$$

Given the property

$$\mathbf{n}^D : \mathbf{n}^D = \frac{3}{2}$$

the *effective creep strain rate* equals the absolute value of the user input

$$\dot{\epsilon}_{\text{cre}} = \sqrt{\frac{2}{3} \dot{\boldsymbol{\epsilon}}_{\text{cr}} : \dot{\boldsymbol{\epsilon}}_{\text{cr}}}$$

Deviatoric creep is very popular to model creep in metals and alloys. For example, Norton's law is a deviatoric creep model.



The effective creep strain and the effective 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 $\dot{\boldsymbol{\epsilon}}_{\text{cr}}$.

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 effective stress, σ_e :

$$\dot{\boldsymbol{\epsilon}}_{\text{cr}} \propto \sigma_e^n$$

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

$$\dot{\epsilon}_{\text{cr}} = A \left(\frac{\sigma_e}{\sigma_{\text{ref}}} \right)^n e^{-\frac{Q}{RT}} \mathbf{n}^D \quad (3-45)$$

Here, A is the creep rate coefficient (SI unit: 1/s), n is the stress exponent (dimensionless), σ_{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 3-44](#).



- 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.

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 effective 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

$$\dot{\boldsymbol{\varepsilon}}_{\text{cr}} = F_{\text{cr}} m \left(\frac{t + t_{\text{shift}}}{t_{\text{ref}}} \right)^{m-1} \mathbf{n}^D \quad (3-46)$$

where \mathbf{n}^D is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-44, and F_{cr} is expressed as in the Norton model:

$$F_{\text{cr}} = A \left(\frac{\sigma_e}{\sigma_{\text{ref}}} \right)^n e^{-\frac{Q}{RT}} \quad (3-47)$$

Here, A is the creep rate coefficient (SI unit: 1/s), n is the stress exponent (dimensionless), σ_{ref} is a reference stress level (SI unit: Pa), t_{ref} and t_{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

$$\dot{\boldsymbol{\varepsilon}}_{\text{cr}} = F_{\text{cr}} m \left(\frac{\varepsilon_{\text{cr,e}} + \varepsilon_{\text{shift}}}{t_{\text{ref}} F_{\text{cr}}} \right)^{\frac{m-1}{m}} \mathbf{n}^D \quad (3-48)$$

where $\varepsilon_{\text{cr,e}}$ is the effective creep strain, and $\varepsilon_{\text{shift}}$ is the effective creep strain shift.

The time and frequency shifts in Equation 3-46 and Equation 3-48 serve two purposes:

- They can be used to initialize a study where some hardening has already taken place.
- The strain rate expressions actually predicts 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 effective 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 (Norton law) for $\alpha\sigma_e < 0.8$ and approaches exponential creep for $\alpha\sigma_e > 1.2$, where $1/\alpha$ is a reference effective 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 σ_{ref} a reference effective stress level (SI unit: Pa). \mathbf{n}^D is a deviatoric tensor coaxial to the stress tensor as defined in [Equation 3-44](#).



- 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 3-44](#).



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 ionic diffusivity along grain boundaries D_{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 size 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 3-44](#). 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 3-44, and the creep rate is calculated from

$$\dot{\mathbf{F}}_{\text{cr}} = A e^{-Q/RT} \left[\sinh \left(\xi \frac{\sigma_{\text{e}}}{s_{\text{a}}} \right) \right]^{\frac{1}{m}}$$

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, ξ 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

$$\dot{s}_{\text{a}} = h_0 \left| 1 - \frac{s_{\text{a}}}{s_{\text{a}}^*} \right|^{a-1} \left(1 - \frac{s_{\text{a}}}{s_{\text{a}}^*} \right) \dot{\mathbf{F}}_{\text{cr}}$$

with the initial condition $s_{\text{a}}(0) = s_{\text{init}}$. 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

$$s_{\text{a}}^* = s_0 \left(\frac{\dot{\mathbf{F}}_{\text{cr}}}{A} e^{-Q/RT} \right)^n$$

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

$$\dot{\boldsymbol{\varepsilon}}_{\text{vp}} = A \left\langle \frac{\mathbf{F}_{\text{y}}}{\sigma_{\text{ref}}} \right\rangle^n \mathbf{n}^D$$

Here, A is the viscoplastic rate coefficient (SI unit: 1/s), n is the stress exponent (dimensionless), σ_{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

$$\mathbf{F}_{\text{y}} = \phi(\boldsymbol{\sigma}) - \sigma_{\text{ys}}$$

The effective stress $\phi(\sigma)$ is either the von Mises, Tresca, or Hill stress; or a user defined expression, and σ_{ys} is the yield stress (which may include a linear or nonlinear [Isotropic Hardening](#) model). The stress tensor used in the effective stress $\phi(\sigma)$ is shifted by what is usually called the *back stress*, σ_{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 effective 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

$$\mathbf{n}^D = \frac{3 \operatorname{dev}(\sigma)}{2 \sigma_{mises}} \quad (3-49)$$

Given the property

$$\mathbf{n}^D : \mathbf{n}^D = \frac{3}{2}$$

the effective 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: W/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_c (SI unit: J/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.Wvpdr`. The dissipated energy density due to creep is available under the variable `solid.Wc` and due to viscoplasticity under the variable `solid.Wvp`. Here `solid` denotes the name of the physics interface node.

CONVERTING BETWEEN DIFFERENT CREEP DATA REPRESENTATIONS

The equation forms described in 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 σ_{ref} and reference time t_{ref} . These values are in a sense superfluous, and can in principle be chosen arbitrarily. The choice of reference values will however affect 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_N and n in the equation

$$\dot{\boldsymbol{\epsilon}}_{\text{cr}} = A_N \boldsymbol{\sigma}_e^n$$

The coefficient A_N has a physical dimension which depends on the value of n and the unit have an implicit dependence on the stress and time units. Converting the data to the form used in COMSOL Multiphysics (Equation 3-45) requires the introduction of the reference stress σ_{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, σ_{c7} is the stress at which the creep strain rate is $10^{-7}/\text{h}$. Data on this form is also easy to enter: You set the reference stress to the value of σ_{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:

- $\sigma_{c7} = 70 \text{ MPa}$, and stress exponent $n = 4.5$.
- $A_N = 4.98 \cdot 10^{-16}$ with respect to units MPa and hours, and stress exponent $n = 4.5$.

In the first case, enter:

- σ_{ref} as $70[\text{MPa}]$
- n as 4.5
- A as $1\text{e-}7[1/\text{h}]$.

In the second case, enter:

- σ_{ref} as $1[\text{MPa}]$
- n as 4.5
- A as $4.98\text{e-}16[1/\text{h}]$

These two sets of data describe the same material.

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

$$\varepsilon_{\text{cr}} = A_{\text{NB}} \sigma_e^{n_t} t^m$$

In this case, the coefficient A_{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 3-46 and Equation 3-47) requires the introduction of an both an arbitrary reference stress σ_{ref} and an arbitrary reference time t_{ref} . If you use the implicit units for which A_{NB} is given as the reference values, then the constant A will have the same numerical value as A_{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

$$\dot{\epsilon}_{cr} \propto (\sinh(\alpha\sigma_e))^n$$

Comparing with the expression in COMSOL Multiphysics,

$$\dot{\epsilon}_{cr} \propto \left(\sinh\left(\frac{\sigma_e}{\sigma_{ref}}\right) \right)^n$$

it is evident that the reference stress should be chosen as

$$\sigma_{ref} = \frac{1}{\alpha}$$

In this case, there is no arbitrariness in the choice of σ_{ref} , since α is an actual material parameter.

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, ξ_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 ξ_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 application 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 ϵ and the temperature T . An additional internal variable is used to compute the *transformation strain* tensor ϵ_{tr} (Ref. 18-19)

$$\psi(\epsilon, T, \epsilon_{tr}) = \rho c T (1 - \ln(T)) + \frac{K}{2} \epsilon_{vol}^2 + G(\epsilon - \epsilon_{tr})^2 + \frac{H_k}{2} (\epsilon_{tr})^2 + f(T) |\epsilon_{tr}| + I(\epsilon_{tr})$$

here, c is the heat capacity at constant pressure, K and G are the bulk and shear moduli, ϵ_{vol} is the volumetric strain, H_k is the hardening modulus, and $I(\epsilon_{tr})$ is the *indicator function* for the strain limit constraint.

The indicator function is defined by

$$I(\epsilon_{tr}) = \begin{cases} 0 & \text{for } |\epsilon_{tr}| < \epsilon_{tr, \max} \\ \infty & \text{otherwise} \end{cases}$$

The term $\psi_{ch} = f(T) |\epsilon_{tr}|$ in the free energy density 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, define from the slope of the limit curve β and the martensite finish temperature M_f

$$f(T) = \beta \langle T - M_f \rangle$$

Here, the operator $\langle . \rangle$ denotes the Macaulay brackets.

The conjugated thermodynamic stress σ_{tr} associated to the transformation strain variable is

$$\sigma_{\text{tr}} = \frac{\partial \Psi}{\partial \varepsilon_{\text{tr}}} = \text{dev}(\sigma) - H \varepsilon_{\text{tr}} - f(T) \frac{\partial |\varepsilon_{\text{tr}}|}{\partial \varepsilon_{\text{tr}}} - \frac{\partial I}{\partial \varepsilon_{\text{tr}}}$$

The evolution of the transformation strain ε_{tr} is given by the so-called *limit function*, which takes the same form as the yield function for metal plasticity.

$$F = |\sigma_{\text{tr}}| - \sigma_{\text{ys}0}$$

The evolution equation for the transformation strain ε_{tr} is computed from the flow rule

$$\dot{\varepsilon}_{\text{tr}} = \lambda_{\text{p}} \frac{\partial F}{\partial \sigma_{\text{tr}}}$$

where the plastic multiplier λ_{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 σ and the temperature field T . Additional internal variables are used to compute the transformation strain tensor ε_{tr} and the martensite volume fraction ξ ([Ref. 20](#))

$$G(\sigma, T, \varepsilon_{\text{tr}}, \xi) = c \left((T - T_0) - T \ln \left(\frac{T}{T_0} \right) \right) - s_0 T - \frac{1}{\rho} \sigma : S : \sigma - \frac{1}{\rho} \sigma : (\varepsilon_{\text{th}} - \varepsilon_{\text{tr}}) + u_0 + \frac{1}{\rho} f(\xi)$$

Here, c is the heat capacity at constant pressure, s_0 is the specific entropy at reference state, S is the compliance matrix, ε_{th} is the thermal strain tensor, ε_{tr} is the transformation strain tensor, u_0 is the specific internal energy at reference state, and $f(\xi)$ is the *transformation hardening function*. The compliance matrix S , is obtained by averaging 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 ε_{tr} is computed from the flow rule

$$\dot{\varepsilon}_{\text{tr}} = \xi \Lambda$$

where the normalized *transformation tensor* Λ changes principal directions depending on the direction of the martensitic transformation.

The variable for the direction of the martensitic transformation, ξ , is calculated from the previous state to determine the expected increment or decrement of the transformation strain tensor ϵ_{tr} . This calculation is computational expensive and it can lead to convergence issues. Since in many applications the transformation direction ξ 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 μ_1 and μ_2 from

$$\mu_1 = \frac{1}{2}\rho\Delta s_0(M_s + A_f) - \rho\Delta u_0 \text{ and } \mu_2 = \frac{1}{4}\rho\Delta s_0(A_s - A_f - M_f + M_s)$$

The smooth hardening function is defined with four smoothing parameters $n_1, n_2, n_3,$ and n_4

$$f(\xi) = \begin{cases} \frac{1}{2}\rho b_M \left(\xi + \frac{\xi^{n_1+1}}{n_1+1} + \frac{(1-\xi)^{n_2+1}}{n_2+1} \right) & \text{for } \xi > 0 \\ \frac{1}{2}\rho b_A \left(\xi + \frac{\xi^{n_3+1}}{n_3+1} + \frac{(1-\xi)^{n_4+1}}{n_4+1} \right) & \text{for } \xi < 0 \end{cases}$$

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 ϵ (instead of \mathbf{S}) and the stresses are denoted by either σ or 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

$$\begin{aligned}\boldsymbol{\sigma} &= c_E \boldsymbol{\varepsilon} - e^T \mathbf{E} \\ \mathbf{D} &= e \boldsymbol{\varepsilon} + \varepsilon_0 \varepsilon_{rS} \mathbf{E}\end{aligned}\tag{3-50}$$

The Stress-Charge form is always used in the variational formulation (weak equation form) which COMSOL Multiphysics uses for discretization and computation.

STRAIN-CHARGE

$$\begin{aligned}\boldsymbol{\varepsilon} &= s_E \boldsymbol{\sigma} + d^T \mathbf{E} \\ \mathbf{D} &= d \boldsymbol{\sigma} + \varepsilon_0 \varepsilon_{rT} \mathbf{E}\end{aligned}\tag{3-51}$$

Most material data appears 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:

$$\begin{aligned}c_E &= s_E^{-1} \\ e &= d s_E^{-1} \\ \varepsilon_S &= \varepsilon_0 \varepsilon_{rS} = \varepsilon_0 \varepsilon_{rT} - d s_E^{-1} d^T\end{aligned}\tag{3-52}$$

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 Piezoelectric Devices 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 3-78](#) with the charge conservation equation of Electrostatics,

$$\nabla \cdot \mathbf{D} = \rho_V\tag{3-53}$$

where the ρ_V is the electric charge concentration. The electric field is computed from the electric potential V as

$$\mathbf{E} = -\nabla V$$

In both Equation 3-53 and Equation 3-78, the constitutive relations Equation 3-50 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 case of geometric linearity, the governing equations are linear and have the following time-harmonic wave solutions:

$$\begin{aligned}\mathbf{u} &= \hat{\mathbf{u}} \exp[i\mathbf{k}(\mathbf{r} + \mathbf{c}t)] \\ V &= \hat{V} \exp[i\mathbf{k}(\mathbf{r} + \mathbf{c}t)]\end{aligned}$$

where \mathbf{k} is the wave number vector that determines the direction of the wave propagation, and \mathbf{c} is the phase velocity (or wave speed).

The expressions for the wave speed can be computed analytically for waves of different 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 pressure wave speed is given by

$$c_{pX} = \frac{1}{\rho} \left[c_{E, 11} + \frac{e_{X1}^2}{\epsilon_0 \epsilon_{rS, XX}} \right]^{1/2}$$

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 wave speed is computed as

$$c_{sXY} = \frac{1}{\rho} \left[c_{E,66} + \frac{e_{X6}^2}{\epsilon_0 \epsilon_{rS,XX}} \right]^{1/2}$$

COMSOL Multiphysics provides predefined variables for the waves speeds for waves of different types and polarizations propagating in the X , Y , and Z directions.

PIEZOELECTRIC DISSIPATION

In order to define dissipation in the piezoelectric material for a time-harmonic analysis, all material properties in the constitutive relations can be complex-valued matrices where the imaginary part defines the dissipative function of the material.

Complex-valued data can be defined directly in the fields for the material properties, or a real-valued material X and a set of loss factors η_X can be defined, which together form the complex-valued material data

$$\tilde{X} = X(1 \pm j\eta_X)$$

The sign in the complex damping terms are defined as

$$\begin{aligned}
\tilde{c}_E &= (1 + i\eta_{c_E})c_E \\
\tilde{s}_E &= (1 - i\eta_{s_E})s_E \\
\tilde{\epsilon}_{rS} &= (1 - i\eta_{\epsilon_{rS}})\epsilon_{rS} \\
\tilde{\epsilon}_{rT} &= (1 - i\eta_{\epsilon_{rT}})\epsilon_{rT} \\
\tilde{e} &= (1 + i\eta_e)e \\
\tilde{d} &= (1 + i\eta_d)d
\end{aligned}$$

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 η_X is positive in order for the material to be thermodynamically stable with the chosen signs.

The sign of the coupling losses requires more considerations, and the chosen sign must be considered as a definition. All values of η_e and η_d does not necessarily have to be positive. For some simple theoretical cases with isotropic loss factors, it can however be shown that the definition above is reasonable:

- Consider a material which in strain-charge form only has mechanical damping. Thus, d is real, and the second transformation law in [Equation 3-52](#) shows that e must have a positive loss factor.
- Consider a material with only coupling loss. Then the second transformation law in [Equation 3-52](#) shows that d and e must have the same loss factor multiplier. From the third transformation law it can be inferred that η_e and η_d in this case must be positive in order for the material to have an appropriate phase lag between \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, σ . 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.



INITIAL STRESS, STRAIN, AND ELECTRIC DISPLACEMENT

Using the functionality available under the Piezoelectric Material feature and Solid Mechanics interface, one can define initial stress (S_0), initial strain (ϵ_0), and remanent electric displacement (\mathbf{D}_r) for models. In the constitutive relation for piezoelectric material these additions appear in the stress-charge formulation:

$$\begin{aligned}\sigma &= c_E(\epsilon - \epsilon_0) - e^T \mathbf{E} + \sigma_0 \\ \mathbf{D} &= e(\epsilon - \epsilon_0) + \epsilon_{0, \text{vac}} \epsilon_{rS} \mathbf{E} + \mathbf{D}_r\end{aligned}$$

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. Thus use these initial field to add, for example, thermal expansion or pyroelectric effects to models.

Magnetostrictive Material

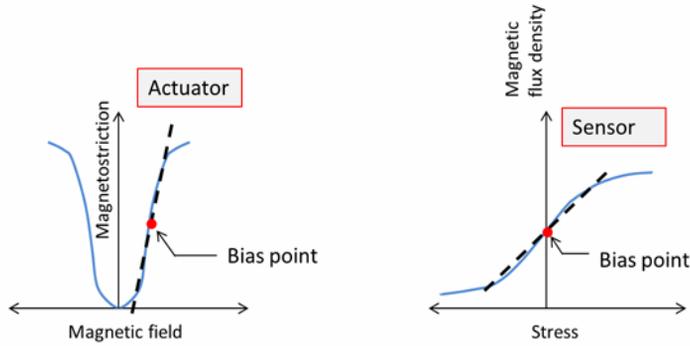
MAGNETOSTRICTION

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

$$\mathbf{S} = c_{\mathbf{H}}\boldsymbol{\varepsilon} - e_{\mathbf{HS}}^T \mathbf{H}$$

$$\mathbf{B} = e_{\mathbf{HS}}\boldsymbol{\varepsilon} + \mu_0 \mu_{\mathbf{rS}} \mathbf{H}$$

Strain-Magnetization

$$\boldsymbol{\varepsilon} = s_{\mathbf{H}}\mathbf{S} + d_{\mathbf{HT}}^T \mathbf{H}$$

$$\mathbf{B} = d_{\mathbf{HT}}\mathbf{S} + \mu_0 \mu_{\mathbf{rT}} \mathbf{H}$$

where μ_0 is the magnetic permeability of free space, $c_{\mathbf{H}}$ and $s_{\mathbf{H}}$ are respectively the stiffness and compliance matrices measured at constant magnetic field, 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:

$$\begin{aligned}c_{\text{H}} &= s_{\text{H}}^{-1} \\e_{\text{HS}} &= d_{\text{HT}}s_{\text{H}}^{-1} \\\mu_{\text{rS}} &= \mu_{\text{rT}} - \frac{1}{\mu_0}d_{\text{HT}}s_{\text{E}}^{-1}d_{\text{HT}}^T\end{aligned}$$

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 [Orthotropic and Anisotropic Materials](#) section.

For a crystalline material with tetragonal symmetry, the strain-magnetization form of the constitutive relations is the following:

$$\begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & 0 & 0 & 0 \\ s_{12} & s_{11} & s_{13} & 0 & 0 & 0 \\ s_{13} & s_{13} & s_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(s_{11} - s_{12}) \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{23} \\ S_{13} \\ S_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & d_{33} \\ 0 & d_{15} & 0 \\ d_{15} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}$$

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{15} & 0 & 0 \\ -\frac{1}{2}d_{33} & -\frac{1}{2}d_{33} & d_{33} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{23} \\ S_{13} \\ S_{12} \end{bmatrix} + \mu_0 \begin{bmatrix} \mu_{11} & 0 & 0 \\ 0 & \mu_{11} & 0 \\ 0 & 0 & \mu_{33} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}$$

The following material data corresponds to Terfenol-D at 100 kA/m bias and 30 MPa prestress ([Ref. 5](#)):

$$s_H = 10^{-11} \cdot \begin{bmatrix} 4.4 & -1.1 & -1.65 & 0 & 0 & 0 \\ -1.1 & 4.4 & -1.65 & 0 & 0 & 0 \\ -1.65 & -1.65 & 3.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 \end{bmatrix} [1/\text{Pa}]$$

$$d_{HT} = 10^{-9} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 16.5 & 0 \\ 0 & 0 & 0 & 16.5 & 0 & 0 \\ -4.3 & -4.3 & 8.6 & 0 & 0 & 0 \end{bmatrix} [\text{m}/\text{A}] \quad \mu_{rT} = \begin{bmatrix} 8.1 & 0 & 0 \\ 0 & 8.1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

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:

$$\varepsilon_{me} = \frac{3}{2} \left[\lambda_{100} \left(\mathbf{m} \otimes \mathbf{m} - \frac{1}{3} I \right) + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} m_i m_j (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \quad (3-54)$$

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 β_i can be written as

$$\lambda = \frac{\Delta l}{l} = \sum_{i,j} \beta_i \beta_j \varepsilon_{ij}$$



The notation lambda is used for strain in this section, which is typical for micromagnetics literature. This should not be mistaken with the same notation used for stretch in other parts of the theory in COMSOL Multiphysics documentation.

Using Equation 3-54, one gets

$$\lambda = \frac{3}{2} \lambda_{100} \left(m_1^2 \beta_1^2 + m_2^2 \beta_2^2 + m_3^2 \beta_3^2 - \frac{1}{3} \right) + 3 \lambda_{111} (m_1 m_2 \beta_1 \beta_2 + m_2 m_3 \beta_2 \beta_3 + m_1 m_3 \beta_1 \beta_3) \quad (3-55)$$

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 3-54 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:

$$\varepsilon_{me} = \frac{3}{2} \frac{1}{M_s^2} \left[\lambda_{100} \text{dev}(\mathbf{M} \otimes \mathbf{M}) + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} M_i M_j (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \quad (3-56)$$

The strain field is deviatoric, and Equation 3-56 exhibits the same properties as Equation 3-54 at saturation, that is, when $|\mathbf{M}| \rightarrow M_s$. Equation 3-55 is replaced by

$$\lambda = \frac{3\lambda_{100}}{2} \frac{1}{M_s^2} \left[M_1^2 \left(\beta_1^2 - \frac{1}{3} \right) + M_2^2 \left(\beta_2^2 - \frac{1}{3} \right) + M_3^2 \left(\beta_3^2 - \frac{1}{3} \right) \right] + 3 \frac{\lambda_{111}}{M_s^2} (M_1 M_2 \beta_1 \beta_2 + M_2 M_3 \beta_2 \beta_3 + M_1 M_3 \beta_1 \beta_3)$$

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):

$$\varepsilon_{me} = \frac{3}{2} \frac{\lambda_s}{M_s^2} \text{dev}(\mathbf{M} \otimes \mathbf{M}) \quad (3-57)$$

The stress in the magnetostrictive material is modeled as

$$\mathbf{S} = \mathbf{c}_H [\varepsilon - \varepsilon_{me}(\mathbf{M})]$$

For isotropic materials, the stiffness matrix \mathbf{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 3-57, 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

Nonlinear magnetization in the magnetostrictive material is found from the following nonlinear implicit relation (Ref. 4):

$$\mathbf{M} = M_s L(|\mathbf{H}_{\text{eff}}|) \frac{\mathbf{H}_{\text{eff}}}{|\mathbf{H}_{\text{eff}}|}$$

where L is the Langevin function

$$L = \coth\left(\frac{3\chi_m |\mathbf{H}_{\text{eff}}|}{M_s}\right) - \frac{M_s}{3\chi_m |\mathbf{H}_{\text{eff}}|}$$

with χ_m being the magnetic susceptibility in the initial linear region.

Other possible choices of the L function are a hyperbolic tangent

$$L = \tanh\left(\frac{\chi_m |\mathbf{H}_{\text{eff}}|}{M_s}\right)$$

and a linear function

$$L = \frac{\chi_m |\mathbf{H}_{\text{eff}}|}{M_s}$$

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

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3}{\mu_0 M_s^2} \left[\lambda_{100} \mathbf{S}_{\text{ed}} + (\lambda_{111} - \lambda_{100}) \sum_{i \neq j} (S_{\text{ed}})_{ij} (\mathbf{e}_i \otimes \mathbf{e}_j) \right] \mathbf{M} \quad (3-58)$$

where \mathbf{H} is the applied magnetic field. The second term in Equation 3-58 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

$$\mathbf{S}_{\text{ed}} = \text{dev}(c_{\text{H}} \boldsymbol{\varepsilon})$$

For isotropic materials, the effective magnetic field expression simplifies into

$$\mathbf{H}_{\text{eff}} = \mathbf{H} + \frac{3\lambda_s}{\mu_0 M_s^2} \mathbf{S}_{\text{ed}} \mathbf{M}$$

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.

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 non-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 , respectively. These four parameters are not independent, so an extra equation stating that

$$a^2 + b^2 + c^2 + d^2 = 1$$

is added.

The connection between the quaternion parameters and the rotation matrix R is:

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix}$$

For the geometrically linear case, the quaternion constraint and the rotation matrix definition are reduced to:

$$R = \begin{bmatrix} 1 & -2d & 2c \\ 2d & 1 & -2b \\ -2c & 2b & 1 \end{bmatrix}$$

$$a = 1$$

In 2D, the rotational degree of freedom is the angle of rotation about the z -axis ϕ , and its relation with the rotation matrix R is:

$$R = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For the geometrically linear case, the 2D rotation matrix is reduced to:

$$R = \begin{bmatrix} 1 & -\phi & 0 \\ \phi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Under translation and rotation of a rigid domain, the complete expression for the displacement of any point on the rigid body is given by:

$$\mathbf{u}_d = \mathbf{u} + (R - I) \cdot (\mathbf{X} - \mathbf{X}_M)$$

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 (ω):

$$\dot{\mathbf{u}} = (i\omega)\mathbf{u} \quad \ddot{\mathbf{u}} = (i\omega)\dot{\mathbf{u}} = -\omega^2\mathbf{u}$$

The same is true for the rotation in 2D since the rigid body rotation ϕ is the degree of freedom. The rigid body angular velocity and acceleration can be evaluated by directly taking the time derivatives of ϕ .

In 3D, the situation is different and the total rotation of the rigid domain can be presented as a function of quaternion:

$$\Theta = \frac{2a \cos(a)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

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’ denotes 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 (ϕ), translational velocity ($\frac{\partial \mathbf{u}}{\partial t}$) and angular velocity (ω) 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.

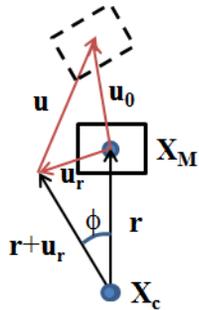


Figure 3-17: Initial displacement of a rigid body

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:

$$\mathbf{u}_c = \mathbf{u} + (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X}_c - \mathbf{X}_M) \quad (3-59)$$

The components of this displacement vector are prescribed individually in the selected coordinate system. Through [Equation 3-59](#), 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 Ω and ϕ_0 are the axis of rotation and angle of rotation respectively.

In 2D, the out-of-plane rotation angle is directly constrained to the prescribed value of the rotation.

LOADS

The loads available for a flexible domain can also be used for a rigid domain. In addition to these boundary conditions, a rigid domain also has global subnodes for applying forces and moments. If you use **Applied Force**, a force and its location can be prescribed in a selected coordinate system. A force implicitly also contributes to the moment unless it is applied at the center of mass of a rigid domain. If an **Applied Moment** node is used, a moment can be prescribed in a selected coordinate system.

CONNECTING TO OTHER BODIES

When a rigid domain and a flexible domain share a boundary (Shell: edge, Beam: point), the connection is automatic. All displacements on the flexible domain are controlled by the degrees of freedom of the rigid domain, so that

$$\mathbf{u}_{\text{flex}} = (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X} - \mathbf{X}_M) + \mathbf{u}$$

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. There are a number of theories to describe such behavior. In the continuum damage mechanics formalism, a *damage* variable represents the amount of deterioration due to crack growth. This damage variable controls the weakening of the material's stiffness, and it produces a nonlinear relation between stress and strain.

For a linear elastic material, Hooke's law relates the *undamaged* stress tensor σ_{un} to the elastic strain tensor:

$$\sigma_{un} = \sigma_{ex} + \mathbf{C} : \varepsilon_{el} = \sigma_{ex} + \mathbf{C} : (\varepsilon - \varepsilon_{incl}) \quad (3-60)$$

here, \mathbf{C} is the fourth order *elasticity tensor*, “:” stands for the double-dot tensor product (or double contraction). The elastic strain ε_{el} is the difference between the total strain ε and all inelastic strains ε_{incl} . There may also be an extra stress contribution σ_{ex} , with contributions from initial, external or viscoelastic stresses.

For the scalar damage models, the *damaged* stress tensor σ_d is computed from the undamaged stress as

$$\sigma_d = (1 - d)\sigma_{un} \quad (3-61)$$

This damaged stress is then used in 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 σ_{un} is used when combining the **Damage** feature with **Creep** or **Viscoelasticity**.

DAMAGE MODELS

The strain-based formulation for the damage model is based on the loading function f such as

$$f = \varepsilon_{eq} - \kappa \leq 0 \quad (3-62)$$

here, ε_{eq} is the equivalent strain, a scalar measure of the elastic strain; and κ is a state variable. The evolution of the state variable κ follows the Kuhn-Tucker loading/unloading conditions

$$f \leq 0, \quad \dot{\kappa} \geq 0, \quad \text{and} \quad \dot{\kappa}f = 0$$

In this formulation, κ is the maximum value of ε_{eq} in the load history. The damage variable d is then computed as a function of the state variable κ and other parameters.

EQUIVALENT STRAIN

Different damage models use different definitions for the equivalent strain ε_{eq} . The **Rankine** damage model defines the equivalent strain from the largest undamaged principal stress σ_{p1} as

$$\varepsilon_{eq} = \frac{\langle \sigma_{p1} \rangle}{E} \quad (3-63)$$

here, the symbol “ $\langle \rangle$ ” are the Macaulay brackets, and E is Young’s modulus. The Macaulay brackets are used since in this formulation only tensile (positive) stresses cause damage.

The **Smooth Rankine** damage model defines the equivalent strain from the three undamaged principal stresses

$$\varepsilon_{eq} = \frac{\sqrt{\langle \sigma_{p1} \rangle^2 + \langle \sigma_{p2} \rangle^2 + \langle \sigma_{p3} \rangle^2}}{E} \quad (3-64)$$

In both the Rankine and the Smooth Rankine damage models, by default only the principal stresses in tension contribute to the damage evolution, but it is also possible to activate damage in compression by including the compressive stresses in the computation of the damaged stress tensor σ_d .

The Euclidean **Norm of the elastic strain tensor** can also be used as a measure for the equivalent strain

$$\varepsilon_{\text{eq}} = \sqrt{\varepsilon_{\text{el}} : \varepsilon_{\text{el}}} \quad (3-65)$$

The Euclidean norm considers both tensile and compressive strains.

For **Mazars damage for concrete**, it is also possible to select from **Mazars** or **Modified Mazars** equivalent strain. For Mazars equivalent strain is defined as

$$\varepsilon_{\text{cq}} = \sqrt{\langle \varepsilon_{\text{el}} \rangle : \langle \varepsilon_{\text{el}} \rangle} \quad (3-66)$$

In the Modified Mazars equivalent strain (Ref. 2 and Ref. 3), a correction factor γ is added to improve the approximation of the failure surface of concrete in multiaxial compression.

$$\varepsilon_{\text{cq}} = \gamma \sqrt{\langle \varepsilon_{\text{el}} \rangle : \langle \varepsilon_{\text{el}} \rangle} \quad (3-67)$$

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

$$\begin{aligned} d(\kappa) &= \left(1 - \frac{\varepsilon_0}{\kappa}\right) \left(\frac{\varepsilon_f}{\varepsilon_f - \varepsilon_0}\right) & \kappa \geq \varepsilon_0, \\ d(\kappa) &= 0 & \kappa < \varepsilon_0 \end{aligned} \quad (3-68)$$

Here, ε_0 denotes the onset of damage, computed from the tensile strength σ_{ts} and Young's modulus E , so that $\varepsilon_0 = \sigma_{\text{ts}}/E$. The parameter ε_f is derived from 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 .

$$\varepsilon_f = \frac{2G_f}{\sigma_{\text{ts}} h_{\text{cb}}} + \frac{\varepsilon_0}{2} \quad (3-69)$$

The **Exponential strain softening** law defines the damage evolution from

$$\begin{aligned} d(\kappa) &= 1 - \frac{\varepsilon_0}{\kappa} \exp\left(-\frac{\kappa - \varepsilon_0}{\varepsilon_f - \varepsilon_0}\right) & \kappa \geq \varepsilon_0 \\ d(\kappa) &= 0 & \kappa < \varepsilon_0 \end{aligned} \quad (3-70)$$

where

$$\varepsilon_f = \frac{G_f}{\sigma_{ts} h_{cb}} + \frac{\varepsilon_0}{2} \quad (3-71)$$

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

$$d(\kappa) = \alpha_t^\beta d_t(\kappa) + \alpha_c^\beta d_c(\kappa) \quad (3-72)$$

here, α_t and α_c are weight functions depending on the current stress state, and β determines the response in shear, i.e. 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**, or the **Mazars damage evolution function**, which is obtained by a combination of linear and exponential strain softening

$$\begin{aligned} d_t(\kappa) &= 1 - (1 - A_t) \frac{\varepsilon_0}{\kappa} - A_t \exp(-B_t(\kappa - \varepsilon_0)) & \kappa \geq \varepsilon_0 \\ d_t(\kappa) &= 0 & \kappa < \varepsilon_0 \end{aligned} \quad (3-73)$$

Here, A_t and B_t are tensile damage evolution parameters, and ε_{0t} is the tensile strain threshold.

The compressive damage evolution law $d_c(\kappa)$ is obtained by the **Mazars damage evolution function**

$$\begin{aligned} d_c(\kappa) &= 1 - (1 - A_c) \frac{\varepsilon_0}{\kappa} - A_c \exp(-B_c(\kappa - \varepsilon_0)) & \kappa \geq \varepsilon_0 \\ d_c(\kappa) &= 0 & \kappa < \varepsilon_0 \end{aligned} \quad (3-74)$$

Here, A_c and B_c are compressive damage evolution parameters, and ϵ_{0c} is the compressive strain threshold. Both the tensile and compressive damage evolution laws can also be specified by **User defined** expressions.

REGULARIZATION

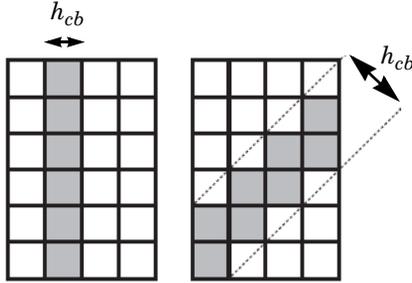
The most common application for the damage models is to describe strain localization, due to for example cracking in quasi-brittle materials. In a damage model without regularization, the deformation during strain softening will always localize in the narrowest possible band, following the principle 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 dissipated energy during softening will decrease upon mesh refinement. The results of a damage model without regularization will therefore be mesh dependent and possibly unstable, hence there is 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, additional equations and variables acting as localization limiters. These methods include non-local averaging of suitable variables, explicit or implicit gradient methods, and phase-field and other variational methods. The following sections describe the two methods available for regularization.

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 in 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 (i.e. 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 band width h_{cb} is defined as:

- $h_{cb} = \sqrt{2a}$, for 2D triangles
- $h_{cb} = \sqrt{a}$, for 2D rectangles
- $h_{cb} = \sqrt[3]{6/\sqrt{2}v}$, 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 3-73](#) and [Equation 3-74](#)) are unaffected by the crack band method.

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* ε_{nl} , through an additional PDE where the equivalent strain ε_{eq} acts as source term. This PDE is solved simultaneously with the displacement field:

$$\varepsilon_{nl} - c \nabla^2 \varepsilon_{nl} = \varepsilon_{eq} \quad (3-75)$$

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)

$$c = \frac{l_{\text{int}}^2}{2n} \quad (3-76)$$

If the fracture energy per unit area is used to define the softening behavior, the size of the damage zone also needs to be entered as an input. This is not necessarily the same as l_{int} .

The strain-based formulation for the damage model (Equation 3-62) is then redefined by the nonlocal equivalent strain ϵ_{nl} instead of the equivalent strain ϵ_{eq}

$$f = \epsilon_{\text{nl}} - \kappa \leq 0 \quad (3-77)$$

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. 13, Ref. 22) proposed a stress-dependent criterion intended at modeling failure in composites. Under 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, σ 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 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 indexes)

$$F_{ii} > 0 \quad \text{and} \quad F_{ii}F_{jj} \geq F_{ij}^2$$

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

$$di = \begin{cases} 1 & g(\sigma) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

here $di = 1$ means damage, and $di = 0$ represents a healthy material.

The *safety factor*, 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 criteria, such as von Mises criterion, $g(\sigma) = \sigma_{mises}/\sigma_{ts} - 1$, and the safety factor is given by $s_f = \sigma_{ts}/\sigma_{mises}$.

The *margin of safety* (Ref. 23) is then computed from the safety factor

$$ms = s_f - 1$$

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](#) or [Nonlinear Elastic Materials](#).

Following Tsai-Wu formalism, different orthotropic criteria can be defined by setting appropriate values for the coefficients in F and f tensors.

ANISOTROPIC TSAI-WU CRITERION

For this criterion, enter twenty one coefficients to define the 6-by-6 matrix F , and six coefficients to define the vector f . The failure criterion is evaluated from the expression

$$g(\sigma) = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^T \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\ F_{12} & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\ F_{13} & F_{23} & F_{33} & F_{34} & F_{35} & F_{36} \\ F_{14} & F_{24} & F_{34} & F_{44} & F_{45} & F_{46} \\ F_{15} & F_{25} & F_{35} & F_{45} & F_{55} & F_{56} \\ F_{16} & F_{26} & F_{36} & F_{46} & F_{56} & F_{66} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}^T \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} - 1$$

here, σ_{ij} are the stress tensor components given in the local coordinate system of the parent node.

ORTHOTROPIC TSAI-WU CRITERION

For this criterion, enter nine coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. 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.

ORTHOTROPIC TSAI-HILL CRITERION

For this criterion, enter six coefficients corresponding to the tensile strengths σ_{tsi} and shear strengths σ_{ssij} given in the local coordinate system of the parent node. 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](#).

ORTHOTROPIC HOFFMAN CRITERION

For this criterion, enter nine coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. 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.

ORTHOTROPIC JENKINS CRITERION

Enter nine coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from

$$g(\sigma) = \max\left(\frac{|\sigma_1|}{\sigma_{s1}}, \frac{|\sigma_2|}{\sigma_{s2}}, \frac{|\sigma_3|}{\sigma_{s3}}, \frac{\sigma_{23}}{\sigma_{ss23}}, \frac{\sigma_{13}}{\sigma_{ss13}}, \frac{\sigma_{12}}{\sigma_{ss12}}\right) - 1$$

here, ϵ_{si} is either the tensile strength or the compressive strength depending whether the stress in the i -direction, σ_i , is positive or negative. The absolute value of the shear stress σ_{ij} in the ij -plane is compared to the corresponding shear strength σ_{ssij} .

ORTHOTROPIC WADDOUPS CRITERION

Waddoups criterion is similar to Jenkins criterion, but the failure criterion is given in terms of strains, not strengths. For this criterion, enter nine coefficients corresponding to the ultimate tensile strains ϵ_{tsi} , ultimate compressive strains ϵ_{csi} , and ultimate shear strains γ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from

$$g(\epsilon) = \max\left(\frac{|\epsilon_1|}{\epsilon_{s1}}, \frac{|\epsilon_2|}{\epsilon_{s2}}, \frac{|\epsilon_3|}{\epsilon_{s3}}, \frac{\gamma_{23}}{\gamma_{ss23}}, \frac{\gamma_{13}}{\gamma_{ss13}}, \frac{\gamma_{12}}{\gamma_{ss12}}\right) - 1$$

here, ϵ_{si} is either the ultimate tensile strain or the ultimate compressive strain depending whether the strain in the i -direction, ϵ_i , is positive or negative. The absolute value of the shear strain γ_{ij} in the ij -plane is compared to the corresponding ultimate shear strain γ_{ssij} .

MODIFIED TSAI-HILL CRITERION

This criterion is derived from Tsai-Wu theory for two-dimensional plane stress problems (Ref. 23). It is available in 2D for the Pate interface and the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from the in plane stresses

$$g(\sigma) = F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 - 1$$

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.

AZZI-TSAI-HILL CRITERION

This criterion is derived from Tsai-Wu theory for two-dimensional plane stress problems. It is available in 2D for the Plate interface and the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from the in plane stresses

$$g(\sigma) = F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 - 1$$

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.

NORRIS CRITERION

This criterion is derived from Tsai-Wu theory for two-dimensional plane stress problems. It is available in 2D for the Plate interface and the Solid Mechanics interface in plane stress, and for the Shell interface in 3D. Enter the coefficients corresponding to the tensile strengths σ_{tsi} , compressive strengths σ_{csi} , and shear strengths σ_{ssij} given in the local coordinate system of the parent node. The failure criterion is then computed from the in plane stresses

$$g(\sigma) = F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 - 1$$

Tsai-Wu coefficients are then computed from

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \quad \text{for } \sigma_{11} > 0, \text{ or } F_{11} = \frac{1}{\sigma_{cs1}^2} \quad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts}^2} \quad \text{for } \sigma_{22} > 0, \text{ or } F_{22} = \frac{1}{\sigma_{cs}^2} \quad \text{for } \sigma_{22} < 0$$

$$F_{66} = \frac{1}{\sigma_{ss12}^2}$$

$$F_{12} = -\frac{1}{2} \sqrt{F_{11} F_{22}}$$

all the other coefficients in F and f tensors are set to zero.

ISOTROPIC VON MISES CRITERION

Von Mises criterion is one of the simplest isotropic criteria to predict yielding in metals and other ductile materials. The failure criterion is computed from the isotropic tensile strength σ_{ts}

$$g(\sigma) = \frac{\sigma_{mises}}{\sigma_{ts}} - 1$$

The effective von Mises stress σ_{mises} is defined from the deviatoric stress tensor, see the section about plasticity and [von Mises Criterion](#). For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

ISOTROPIC TRESCA CRITERION

Tresca criterion is similar to von Mises criterion. The failure criterion is computed from the isotropic tensile strength σ_{ts}

$$g(\sigma) = \frac{\sigma_{tresca}}{\sigma_{ts}} - 1$$

here, Tresca effective stress is defined in terms of principal stresses, $\sigma_{tresca} = \sigma_1 - \sigma_3$, see [Tresca Criterion](#). For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

ISOTROPIC RANKINE CRITERION

St. Venant criterion is similar to Tresca criterion, as the failure criterion is given in terms of principal stresses. For this criterion, enter the tensile strength σ_{ts} , and the compressive strength σ_{cs} . The failure criterion is then computed from

$$g(\sigma) = \max\left(\frac{|\sigma_{p1}|}{\sigma_s}, \frac{|\sigma_{p2}|}{\sigma_s}, \frac{|\sigma_{p3}|}{\sigma_s}\right) - 1$$

here, σ_s is either the tensile strength or the compressive strength depending whether the principal stress, σ_{pi} , is positive or negative. For ductile materials the tensile strength corresponds to the yield stress, while for brittle materials it corresponds to the failure strength.

ISOTROPIC 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 criterion, enter the ultimate tensile strains ϵ_{ts} , and the ultimate compressive strains ϵ_{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, ϵ_s is either the ultimate tensile strain or the ultimate compressive strain depending whether the principal strain, ϵ_{pi} , is positive or negative. For ductile materials the ultimate tensile strain corresponds to the strain at yielding, while for brittle materials it corresponds to the strain at failure.

MOHR-COULOMB CRITERION

The Mohr-Coulomb criterion is similar to the Tresca criterion, as the failure criterion is given in terms of principal stresses, see [Mohr-Coulomb Criterion](#) for soil plasticity. For this criterion, enter the cohesion c , and the angle of internal friction ϕ . The failure criterion is then computed from

$$g(\sigma) = \frac{\sqrt{J_2}m(\theta)}{k - \alpha I_1} - 1$$

and the failure index from

$$fi = \frac{\sqrt{J_2}m(\theta)}{k - \alpha I_1}$$

where

$$m(\theta) = \sqrt{\frac{1}{3}} \left((1 + \sin \phi) \cos \theta - (1 - \sin \phi) \cos \left(\theta + \frac{2\pi}{3} \right) \right), \quad \alpha = \sin \phi / 3, \text{ and}$$

$$k = c \cos \phi$$

The cohesion and the angle of internal friction are related to the tensile and compressive strengths by the expressions

$$\sin(\phi) = \frac{\sigma_{cs1} - \sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}} \quad \text{and} \quad c \cos(\phi) = \frac{\sigma_{cs1} \sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}}$$

DRUCKER-PRAGER CRITERION

The Drucker-Prager criterion approximates the Mohr-Coulomb criterion by a smooth function (a cone in the stress space), see [Drucker-Prager Criterion](#) for soil plasticity.

The failure criterion is computed from the stress invariants I_1 and J_2 , and two material parameters, α and k ,

$$g(\sigma) = \frac{\sqrt{J_2}}{k - \alpha I_1} - 1$$

The material parameters α and k are related to the cohesion c and angle of internal friction ϕ in the Mohr-Coulomb criterion, see [Drucker-Prager Criterion](#) for details.

Also, the cohesion and the angle of internal friction can be related to the tensile and compressive strengths, see [Mohr-Coulomb Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{k - \alpha I_1}$$

BRESLER-PISTER CRITERION

The Bresler-Pister criterion 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. The failure criterion is computed from the stress invariants I_1 and J_2 , and three parameters, k_1 , k_2 , and k_3 ,

$$g(\sigma) = \frac{\sqrt{J_2}}{-k_1 I_1^2 - k_2 I_1 - k_3} - 1$$

The parameters k_1 , k_2 , and k_3 are computed from the uniaxial compressive strength σ_c , the uniaxial tensile strength σ_t , and the biaxial compression strength σ_b , see [Bresler-Pister Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{-k_1 I_1^2 - k_2 I_1 - k_3}$$

WILLAM-WARNKE CRITERION

The Willam-Warnke 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 θ , 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, σ_c is the uniaxial compressive strength, σ_t is uniaxial tensile strength, and σ_b is the biaxial compressive strength. The function $r(\theta)$ describes the segment of an ellipse on the octahedral plane, see [Willam-Warnke Criterion](#) for details. The failure index is computed from

$$fi = \frac{\sqrt{J_2}}{\sqrt{\frac{5}{2}} r(\theta) \sigma_c \left(1 + \left(\frac{1}{\sigma_b} - \frac{1}{\sigma_t}\right) \frac{I_1}{3}\right)}$$

OTTOSSEN CRITERION

The Ottosen criterion is a five-parameter failure criterion proposed for short-time loading of concrete. It corresponds to a smooth convex failure surface with curved meridians

$$g(\sigma) = \frac{a J_2}{\sigma_c (\sigma_c - b I_1)} + \frac{\lambda(\theta) \sqrt{J_2}}{\sigma_c - b I_1} - 1$$

In this formulation, the parameters a and b are positive and dimensionless, and σ_c is the uniaxial compressive strength for concrete (also with a positive sign). The dimensionless function $\lambda(\theta)$ depends on the Lode angle θ and two positive parameters k_1 and k_2 , see [Ottosen Criterion](#) for details. The failure index is computed from

$$f_i = \frac{aJ_2}{\sigma_c(\sigma_c - bI_1)} + \frac{\lambda(\theta)\sqrt{J_2}}{\sigma_c - bI_1}$$

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.

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 ρ is the current mass density. For the material frame formulation used in COMSOL Multiphysics, it is more appropriate to use a Lagrangian version if the equation:

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_X P^T + \mathbf{F}_V$$

Now that the first Piola-Kirchhoff stress tensor, P , is used. \mathbf{F}_V is a body force with components in the current configuration but given with respect to the undeformed volume, and ρ_0 is the initial mass density. Note the gradient operators are not the same: in the first case the gradient is taken with respect to the spatial coordinates, and in the second case with respect to the material coordinates. Using the more common second Piola-Kirchhoff stress tensor, S , the same equation reads

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla_X (F S) + \mathbf{F}_V \quad (3-78)$$

where F is the deformation gradient. The COMSOL Multiphysics implementation of the equations in the Solid Mechanics interface is however not based on the equation of motion directly, but rather on the *principle of virtual work*.

The principle of virtual work states that the sum of the internal virtual work and the external virtual work are equal. The internal virtual work is the work done by the current stress state on a kinematically admissible variation in strains. The external

virtual work is the work done by all forces (acting on domains, boundaries, edges, or points) when multiplied with the variation in displacements corresponding to the variation in strains. The virtual displacements $\delta \mathbf{u}$ are in the finite element formulation represented by the `test()` operator in COMSOL Multiphysics. For a stationary case, the virtual work δW is written as

$$\delta W = \int_V (-\delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} + \delta \mathbf{u} \cdot \mathbf{F}_V) dv + \int_S (\delta \mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta \mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta \mathbf{u} \cdot \mathbf{F}_p)$$

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.

$$\delta W = \int_V (-\delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} + \delta \mathbf{u} \cdot \mathbf{F}_V - \rho \delta \mathbf{u} \cdot \mathbf{u}_{tt}) dv + \int_S (\delta \mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\delta \mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\delta \mathbf{u} \cdot \mathbf{F}_p) \quad (3-79)$$

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 strain $\boldsymbol{\varepsilon}$ is the Green-Lagrange strain and the stress $\boldsymbol{\sigma}$ is the second Piola-Kirchhoff stress.

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

Frequency-Domain Studies

In the frequency domain, the frequency response is studied when applying harmonic loads. Harmonic loads are specified using two components:

- The amplitude value in direction m , F_m
- The corresponding phase, ϕ_m

To derive the equations for the linear response from harmonic excitation loads

$$F_{m, \text{freq}} = F_m(f) \cos(\omega t + \phi_m)$$
$$\mathbf{F}_{\text{freq}} = \begin{bmatrix} F_{x, \text{freq}} \\ F_{y, \text{freq}} \\ F_{z, \text{freq}} \end{bmatrix}$$

Assume a harmonic response with the same angular frequency as the excitation load

$$u = u_{\text{amp}} \cos(\omega t + \phi_u)$$
$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The relationship can also be described using complex notation with

$$u = \text{Re}(u_{\text{amp}} e^{j\phi_u} e^{j\omega t}) = \text{Re}(\tilde{u} e^{j\omega t}) \quad \text{where } \tilde{u} = u_{\text{amp}} e^{j\phi_u}$$

$$\mathbf{u} = \text{Re}(\tilde{\mathbf{u}} e^{j\omega t})$$

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 λ is not necessarily a purely imaginary number. Any damping included in the problem will automatically cause the eigenfrequencies to become complex valued.

In addition to the eigenfrequency, the quality factor, Q , and decay factor, δ , for the model can be examined:

$$Q = \frac{\text{Im}(\lambda)}{2\text{Re}(\lambda)}$$

$$\delta = \text{Re}(\lambda)$$

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 ϕ_i is the eigenmode in terms of a vector of degrees of freedom. The modal participation factors are defined as

$$\Gamma_{ij} = \frac{1}{m_i} \sum_{n,m} \phi_i^n M_{nm} \psi_j^m$$

where ψ_j^m represent the unit rigid-body modes for translation and rotation.

The effective mass in direction j for mode i can 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 ψ can be represented as columns of the following matrix:

$$\begin{bmatrix} \Psi_{TX} & \Psi_{TY} & \Psi_{TZ} & \Psi_{RX} & \Psi_{RY} & \Psi_{RZ} \\ L & 0 & 0 & 0 & Z - Z_0 & -(Y - Y_0) \\ 0 & L & 0 & -(Z - Z_0) & 0 & X - X_0 \\ 0 & 0 & L & Y - Y_0 & -(X - X_0) & 0 \end{bmatrix}$$

where L represents a unit length.

The translational and rotational participation factors can be computed as, respectively:

$$\Gamma_T = [\Gamma_{TX} \ \Gamma_{TY} \ \Gamma_{TZ}] = \frac{L}{m_F} \int \mathbf{u} dm$$

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.

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 relative damping. 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:

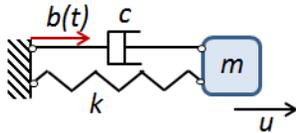
- Single Degree of Freedom System
- Definition of a Response Spectrum
- Solution Using Response Spectrum
- Mode Summation
 - Partitioning Into Periodic and Rigid Modes
 - Gupta Method
 - Lindley-Yow Method
 - Combining Periodic and Rigid Modes
 - Summing the Periodic Modes
 - SRSS Method
 - Grouping 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
 - 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 relative damping is

$$\zeta = \frac{c}{2\sqrt{km}}$$

It can be seen that the support movement acts as a forcing term, and that the solution depends only on two parameters ω_0 and ζ , and not on the individual values of m , c , and k .

Instead of using the absolute displacement as DOF, one can use the relative displacement between the mass and the base,

$$u_r = u - b$$

The equation of motion can then be stated as

$$\ddot{u}_r + 2\zeta\omega_0\dot{u}_r + \omega_0^2 u_r = -\dot{b} \quad (3-80)$$

In practice, such approach means a frame transformation, where the support movement appears as a gravity-like load.

DEFINITION OF A RESPONSE SPECTRUM

For given values of ω_0 , ζ , and $b(t)$, it is a trivial task to solve Equation 3-80 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

$$\begin{aligned} S_v &\approx \omega_0 S_a \\ S_a &\approx \omega_0^2 S_d \end{aligned}$$

Such spectra are called the pseudo-velocity spectrum and the pseudo-acceleration 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

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t) \quad (3-81)$$

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 3-81 can be written as

$$\mathbf{M}(\ddot{\mathbf{u}}_r + \ddot{\mathbf{b}}) + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = \mathbf{0}$$

or

$$\mathbf{M}\ddot{\mathbf{u}}_r + \mathbf{C}\dot{\mathbf{u}}_r + \mathbf{K}\mathbf{u}_r = -\mathbf{M}\ddot{\mathbf{b}}$$

Then, the fact that a rigid body motion does not give any elastic or viscous forces has been used, so that

$$\mathbf{K}\mathbf{b} = \mathbf{0}$$

$$\mathbf{C}\dot{\mathbf{b}} = \mathbf{0}$$

By solving the undamped eigenvalue problem

$$(\mathbf{K} - \omega^2 \mathbf{M})\boldsymbol{\varphi} = \mathbf{0}$$

with the grounded nodes being fixed, a set of eigenmodes that can represent the relative displacements is obtained. Since the mode displacements are zero at the support points, it is however clear that the modes cannot represent the absolute displacements, for which the support points are moving.

By standard operations for mode superposition (and assuming mass matrix normalization and a diagonalizable damping matrix) the decoupled modal equations are

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2 q_j = -\phi_j^T \mathbf{M}\ddot{\mathbf{b}} \quad (3-82)$$

The equation is similar to the standard SDOF system in [Equation 3-80](#).



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 3-82](#), q_j is the modal coordinate for mode j , so that the relative displacement can be written as a superposition of the eigenmodes $\boldsymbol{\varphi}_j$.

$$\mathbf{u}_r = \sum_j \boldsymbol{\varphi}_j q_j = \boldsymbol{\Phi} \mathbf{q}$$

The support motion can be written as

$$\mathbf{b}(t) = b_x(t)\mathbf{1}_x + b_y(t)\mathbf{1}_y + b_z(t)\mathbf{1}_z \quad (3-83)$$

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 3-83](#) in [Equation 3-82](#) gives

$$\ddot{q}_j + 2\zeta\omega_0\dot{q}_j + \omega_0^2 q_j = -\Gamma_{xj}\ddot{b}_x(t) - \Gamma_{yj}\ddot{b}_y(t) - \Gamma_{zj}\ddot{b}_z(t)$$

The multipliers Γ_{kj} are the modal participation factors defined as

$$\Gamma_{kj} = \phi_j^T \mathbf{M} \mathbf{1}_k$$

Thus, the maximum amplitude of mode j , when loaded by a base motion described by a response spectrum in direction k , is

$$\hat{q}_{kj} = S_d(\omega_j, \zeta_j, b_k(t)) \Gamma_{kj} = S_{d,k} \Gamma_{kj}$$

or, using the pseudo-acceleration 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, tress component, effective 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, $R_{pI,j}$ and $R_{rI,j}$ denote the results from an individual eigenmode j .

Not all analyses require a separation into periodic and rigid modes. In such case, all modes are treated as periodic.

Partitioning Into Periodic and Rigid Modes

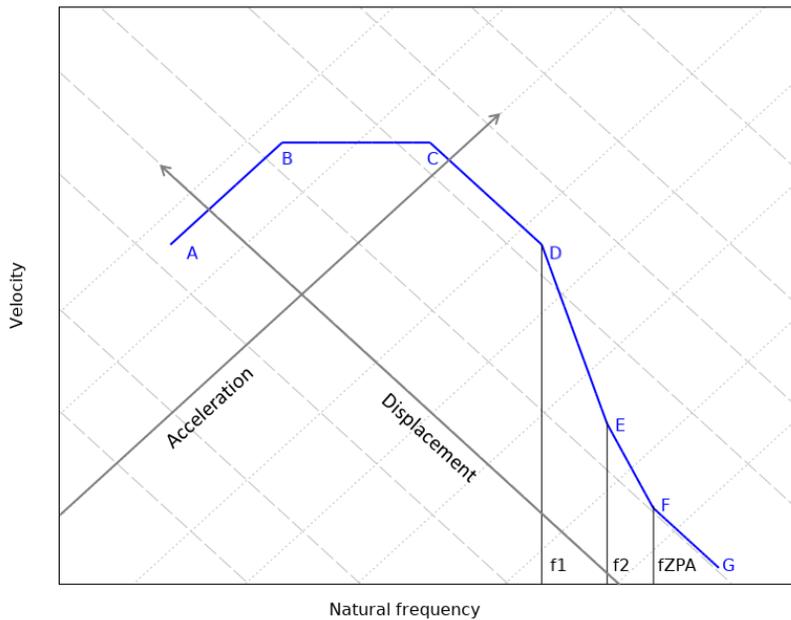


Figure 3-18: A schematic tripartite plot of a design response spectrum. Both axes have logarithmic scales.

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

$$R_{I,j} = R_{pI,j}^2 + R_{rI,j}^2$$

The difference between the two methods lies in how the coefficients α_j are determined. For low frequencies, it should approach the value 0 (fully periodic modes). And for high frequencies, the value is 1 (fully correlated rigid modes).

Gupta Method

In the Gupta method, α_j is a linear function of the logarithm of the natural frequency.

$$\alpha_j = 0 \quad f_j \leq f_1$$

$$\alpha_j = \frac{\log(f_j/f_1)}{\log(f_2/f_1)} \quad f_1 < f_j < f_2$$

$$\alpha_j = 1 \quad f_j \geq f_2$$

where f_1 and f_2 are two key frequencies. Thus, for eigenfrequencies below f_1 , the modes are considered as purely periodic, and above f_2 as purely rigid. In the original Gupta method, the lower key frequency is given by

$$f_1 = \frac{S_{a,\max}}{2\pi S_{v,\max}}$$

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 3-18](#), 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.

Lindley-Yow Method

In the Lindley-Yow method, the coefficient α_j depends directly on the response spectrum values, and not only on the frequency. As a consequence, it is possible that a

certain mode can be considered as having a different degree of rigidness for different excitation directions

$$R_{pI,j} = \frac{\sqrt{S_{a,I}^2(f_j) - S_{ZPA,I}^2}}{S_{a,I}(f_j)} R_{I,j}$$

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 pseudo-acceleration). It also corresponds to the F-G part of the spectrum in [Figure 3-18](#).

Thus,

$$\alpha_{j,I} = \frac{S_{ZPA,I}}{S_{a,I}(f_j)}$$

The value of α_j must be in the range from 0 to 1, and it must increase with frequency. For this reason, NRC RG 1.92 requires that α_j must be set to zero for any eigenmodes below point C in [Figure 3-18](#). For a general spectrum, this is implemented as a strict requirement that α_j has a monotonous decrease with decreasing frequency from f_{ZPA} . As soon as an increase in α_j is found, the value is set to zero for all lower frequencies.

Combining Periodic and Rigid Modes

Once the periodic and rigid responses have been summed up separately, they are combined as

$$R_I = R_{pI}^2 + R_{rI}^2$$

Summing the Periodic Modes

All summation rules for periodic modes except [Absolute Value Sum](#) can be summarized by the following expression:

$$R_{pI}^2 = \sum_{i=1}^{IN} \sum_{j=1}^{IN} C_{ij} R_{pI,i} R_{pI,j}$$

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

$$R_{pI}^2 = \sum_{i=1}^{IV} R_{pI,i}^2 + 2 \sum_{i=1}^{IV} \sum_{j=i+1}^{IV} C_{ij} R_{pI,i} R_{pI,j}$$

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 exists 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 ζ_i is the modal damping. In the implementation in COMSOL Multiphysics, all modes are assumed to have the same damping.

The modified damping, ζ_i' , is defined as

$$\zeta_i' = \zeta_i + \frac{1}{\pi t_d f_i}$$

where t_d is a separate input, called the time of duration. The value of ζ_i differs between modes, even though the damping is constant.

Der Kiureghian Correlation Coefficient (CQC)

This method, which is often called *complete quadratic combination* (CQC), is similar to the previous double sum method. The general expression contains modal damping values. Given that a single damping value is used here, the mode correlation expression can be simplified to

$$C_{ij} = \frac{8\zeta^2 \sqrt{f_i f_j} (f_i + f_j) f_i f_j}{(f_i^2 - f_j^2)^2 + 4f_i f_j (f_i + f_j)^2} \quad (3-84)$$

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 ϕ ,

$$r_k = \phi_k^T \mathbf{f}$$

In the base excitation context when response spectrum analysis is used,

$$\mathbf{f}_I = -\mathbf{M}\ddot{b}_I(t)\mathbf{1}_I$$

so the modal load is

$$r_{k,I} = -\Gamma_{Ik} \ddot{b}_I(t)$$

Thus, the missing mass load is

$$\mathbf{f}_{c,I} = -\mathbf{M} \ddot{b}_I \mathbf{1}_I + \sum_k \Gamma_{Ik} \ddot{b}_I (\mathbf{M} \phi_k)$$

For the rigid body modes, the maximum ground acceleration during the event is equal to the ZPA. The static load is thus

$$\mathbf{f}_{c,I} = S_{ZPA} \mathbf{M} \left(-\mathbf{1}_I + \sum_k \Gamma_{Ik} \phi_k \right) \quad (3-85)$$

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 3-85, 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

$$\sum_k \Gamma_{Ik} \phi_k - \mathbf{1}_I \quad (3-86)$$

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.

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 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

$$R_{pI}^2 = \sum_{i=1}^{IV} \sum_{j=1}^{IV} C_{ij} R_{pI,i} R_{pI,j}$$

where the Der Kiureghian expression Equation 3-84 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:

$$R_{pXY}^2 = \sum_{i=1}^{IV} \sum_{j=1}^{IV} C_{ij} R_{pX,i} R_{pY,j}$$

It is now conceptually assumed that the response spectra are instead applied in a local coordinate system $X'-Y'$ which is rotated an angle θ with respect to the X - Y orientations. It can then be shown that

$$\begin{aligned} R_{pX',i} &= R_{pX,i} \cos \theta + R_{pY,i} \sin \theta \\ R_{pY',i} &= R_{pY,i} \cos \theta - R_{pX,i} \sin \theta \end{aligned}$$

Also, if the relation between the two spectra in the horizontal plane is such that

$$S_Y = \gamma S_X$$

the same ratio γ 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 θ_{\max} giving the maximum response $R(\theta_{\max})$ turns out to be independent of γ , and it has the value

$$\theta_{\max} = \frac{1}{2} \operatorname{atan} \left[\frac{2R_{pXY}^2}{R_{pX}^2 - R_{pY}^2} \right]$$

There are two roots for θ_{\max} , both of which must be checked.

The attractiveness of the CQC3 method is that the same spectrum can be applied to an arbitrary pair of orthogonal axes. The scaling of the secondary spectrum, as well as the orientation of the worst direction, is taken care of by the method.

Extending to rigid modes

As mentioned above, the original CQC3 method only deals with the periodic part of the solution, so it is limited to cases dominated by such modes. It is however possible to make an extension taking also the rigid (high frequency) modes.

Studying how the rigid modes enter the problem when using CQC (or any of the similar combination rules) together with SRSS spatial combination gives some insight:

$$R^2 = \sum_{I=1}^{\sigma} R_I^2 = \sum_{I=1}^{\sigma} (R_{pI}^2 + R_{rI}^2) = \sum_{I=1}^{\sigma} \left(\sum_{i=1}^{IV} \sum_{j=1}^{IV} C_{ij} R_{pI,i} R_{pI,j} + R_{rI}^2 \right)$$

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 θ_{\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 λ 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

$$\delta W = \int_V (-\delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} - \lambda \delta(\boldsymbol{\varepsilon}_{\text{GL}} - \boldsymbol{\varepsilon}) : \boldsymbol{\sigma}_1) dv = 0 \quad (3-87)$$

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}_{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 3-87](#), while the first term uses the linear (engineering) strain tensor.

If, however, geometric nonlinearity is selected in the linear buckling study step, [Equation 3-87](#) 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 α_{dM} and the stiffness damping coefficient β_{dK} . Rayleigh damping will give the following contribution to the virtual work

$$\delta W = \int_V \left(-\delta(\nabla \mathbf{u}) : \beta_{dK} \frac{\partial P}{\partial t} - \alpha_{dM} \rho \delta \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} \right) dv$$

Here P is the first Piola-Kirchhoff stress tensor.

Since Rayleigh damping is added directly to the virtual work equation, it does not affect the constitutive relation. As a consequence, the stresses and strains will for a linear elastic material still be in phase. This stands in contrast to the other damping models.

Rayleigh damping can be used both in the time domain and in the frequency domain.

Rayleigh damping is not directly related to any physical property. Historically, it was introduced since it was numerically attractive to have a damping matrix which was a linear combination of the mass and stiffness matrices

$$C = \alpha M + \beta K$$



This operation is usually implied to be done at the global assembled matrix level. Such an interpretation is however only meaningful for pure structural mechanics problems, but not in a general multiphysics context. For this reason, Rayleigh damping in COMSOL Multiphysics is a material property, rather than a global property of the system of equations. 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 η_s , this means that

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

where D is the constitutive matrix computed from the material data, and D^c is the complex constitutive matrix used when computing the stresses. For a linear elastic material, this would be equivalent to multiplying Young's modulus by the factor $(1 + j\eta_s)$. For a nonlinear elastic material, this applies to the tangential stiffness.

It is also possible to give individual loss factors for each entry in the constitutive matrix, so that

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

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:

$$\begin{aligned}
E_m^c &= (1 + j\eta_{E,m})E_m \\
G_m^c &= (1 + j\eta_{G,m})G_m \\
m &= 1, 2, 3
\end{aligned}$$

The complex moduli are then used to form the constitutive matrix.

For hyperelastic materials, the loss information appears as a multiplier in the strain energy density, and thus in the second Piola-Kirchhoff stress:

$$S = (1 + j\eta_s) \frac{\partial W_s}{\partial E}$$

For loss factor damping, the following definition is used for the elastic part of the entropy:

$$S_{\text{elast}} = \alpha : (s - j\eta_s(\mathbf{C} : \boldsymbol{\varepsilon}))$$

note that S_{elast} here denotes the entropy contribution and not any stress.

This is because the entropy is a function of state and thus independent of the strain rate, while the damping represents the rate-dependent effects in the material (for example, viscous or viscoelastic effects). The internal work of such inelastic forces averaged over the time period $2\pi/\omega$ can be computed as:

$$Q_h = \frac{1}{2} \omega \eta_s \text{Real}(\boldsymbol{\varepsilon} : \text{Conj}(\mathbf{C} : \boldsymbol{\varepsilon}))$$

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 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 η_b and η_v are the bulk and shear viscosity coefficients, respectively.

Viscous damping can be used in both frequency and time domain analyses.

In case of geometric nonlinearity, the viscous stress is treated as being a Cauchy stress acting in the actual configuration (spatial frame). The resulting contribution to the second Piola-Kirchhoff stress is calculated as

$$\mathbf{S}_q = \left(\eta_b - \frac{2}{3} \eta_v \right) \dot{J}_{el} \mathbf{C}_{el}^{-1} + \eta_v \mathbf{C}_{el}^{-T} \dot{\boldsymbol{\varepsilon}}_{el} \mathbf{C}_{el}^{-1}$$

where \mathbf{C}_{el}^{-1} is the inverse of the elastic Cauchy-Green tensor, and J_{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
- Energy Quantities

Distributed Loads

The direction of an explicitly applied distributed load must be given with reference to a local or global coordinate system in the spatial frame, but its magnitude must be with reference to the undeformed reference (or material) area. That is, the relation between the true force \mathbf{f} acting on the current area da and the specified distributed load \mathbf{F} acting on the material area dA is $\mathbf{f} da = \mathbf{F} dA$.

When the solid is subjected to an external pressure, p , the true force on a surface element acts with magnitude p in the current area da in the normal direction \mathbf{n} :

$$\mathbf{f} = p \mathbf{n} da$$

Therefore, the pressure load type specifies the distributed load as

$$\mathbf{F} = p \mathbf{n} \frac{da}{dA}$$

where both the normal \mathbf{n} and area element da are functions of the current displacement field.

Another view of how to interpret the load, is to express it in the first Piola-Kirchhoff stress tensor P via the following formula:

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

where the normal \mathbf{n}_0 corresponds to the undeformed surface element. Such a force vector is often referred to as the *nominal traction*.

Plane Stress

In a plane stress condition, the out-of-plane deformation causes the thickness to change, and this area effect is included explicitly.

The equation transforms to

$$\mathbf{F} = p \mathbf{n} \frac{dl}{dL} \left(1 + \frac{\partial w}{\partial z} \right)$$

where l and L are the current and original in-plane lengths of the boundary.

Axial Symmetry

To account for the radial deformation changing the circumference and therefore the area element, the distributed load is applied as

$$\mathbf{F} = p \mathbf{n} \frac{da}{dA} \frac{(R + u)}{R}$$

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 ρ should be considered as generalized. It can represent mass per unit volume, mass per unit area, mass per unit length, or even mass, depending on the dimensionality of the object giving the contribution.

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 total 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 total 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}_{ax} . The rotation is represented by the angular velocity vector:

$$\boldsymbol{\Omega} = \Omega \mathbf{e}_{\text{ax}}$$

where Ω 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 η is the loss factor and i is the imaginary unit. It is also possible to give individual loss factors for each component in the stiffness matrix \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 “up” and “down” sides 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

$$k_n = \frac{E(1-\nu)}{d_s(1+\nu)(1-\nu)}$$

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 $\mathbf{e}_{\text{aux}} = \mathbf{e}_X$. If the *X* direction cannot be used, the *Y* direction is instead used as the auxiliary direction, $\mathbf{e}_{\text{aux}} = \mathbf{e}_Y$.
- 2 The local *y* direction is obtained from the part of the auxiliary direction which is orthogonal to the normal direction \mathbf{n} :

$$\mathbf{e}_y = \frac{\mathbf{e}_{\text{aux}} - (\mathbf{e}_{\text{aux}} \cdot \mathbf{n})\mathbf{n}}{|\mathbf{e}_{\text{aux}} - (\mathbf{e}_{\text{aux}} \cdot \mathbf{n})\mathbf{n}|}$$

- 3 The local *z* direction is orthogonal to the normal and the local *y* direction:

$$\mathbf{e}_z = \mathbf{n} \times \mathbf{e}_y$$



Springs and Dampers in the [Structural Mechanics Modeling](#) chapter.

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.

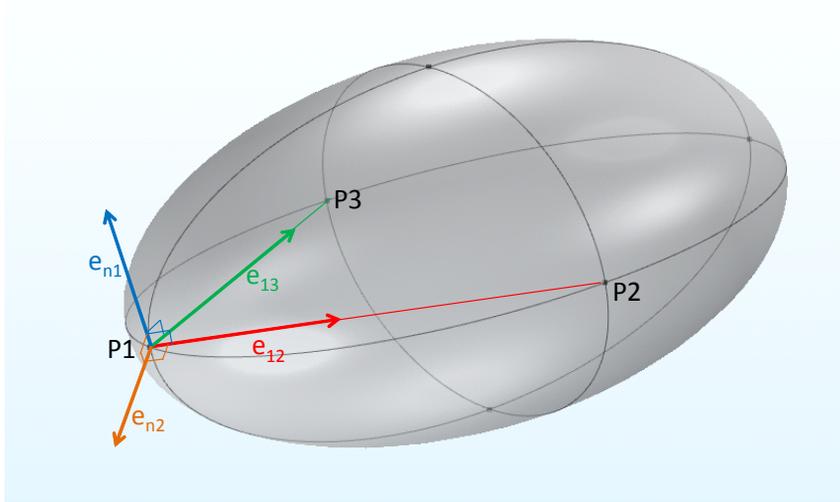


Figure 3-19: Selection of constraint orientations: P1 is fixed, P2 is constrained so that it can only move in the \mathbf{e}_{12} direction, and P3 is constrained in the \mathbf{e}_{n1} direction.

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 φ represents either the periodicity of the eigenmode for an eigenfrequency analysis or the periodicity of the excitation signal in case of a frequency-response analysis. In the latter case, the excitation is typically given as a load vector

$$\mathbf{F} = -\mathbf{F}_0 \exp[-im \operatorname{atan}(Y/X)]$$

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**, the 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 a 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 give by

$$\varepsilon(\mathbf{X}) = \alpha(\mathbf{X}, T)(T - T_{\text{ref}})$$

Note that this is the thermal expansion of the virtual surroundings of the structure being analyzed, so it is unrelated to the thermal strains of the structure itself.

The strain field must be converted into a displacement field $\mathbf{u}(\mathbf{X})$ such that

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

If the strain field fulfills the general compatibility relations, it is in principle possible to integrate the above relation. The procedure is outlined in [Ref. 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 indexes 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. Since 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

$$u_i(\mathbf{X}) = \int_0^1 \left(\varepsilon_{il} + (1-s)p_k \left(\frac{\partial \varepsilon_{il}}{\partial X_k} - \frac{\partial \varepsilon_{kl}}{\partial X_i} \right) \right) p_l ds \quad (3-88)$$

This integral can be computed using the built-in `integrate()` operator as long as the strain field is an explicit function of the material frame coordinates \mathbf{X} .

For the physics interfaces which have rotational degrees of freedom (Beam, Shell and Plate), not only the displacement, but also the rotation of the constraint is needed. For a given displacement field $\mathbf{u}(\mathbf{X})$, the infinitesimal rotation vector Θ is given by

$$\Theta = \frac{1}{2} \nabla \times \mathbf{u}$$

Applying the rotation operator to Equation 3-88 gives

$$\Theta_m(\mathbf{X}) = \frac{1}{2} \varepsilon_{mni} \int_0^1 \left((3-2s) \frac{\partial \varepsilon_{il}}{\partial X_n} + (1-s) p_k \frac{\partial^2 \varepsilon_{il}}{\partial X_k \partial X_n} \right) p_l ds$$

where ε_{mni} is the permutation tensor.

Note that the general compatibility requirements will not be fulfilled for arbitrary expressions for the thermal strain distribution. In such cases, the stresses caused by the

constraints cannot completely be removed, but they will be significantly decreased. The results will then depend on the choice of reference point.



Constraints and Thermal Expansion in the Structural Mechanics Modeling chapter.

Fluid-Structure Interaction 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,

$$\mathbf{f} = \mathbf{n} \cdot \left\{ -p\mathbf{I} + \left(\mu(\nabla\mathbf{u}_{\text{fluid}} + (\nabla\mathbf{u}_{\text{fluid}})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u}_{\text{fluid}})\mathbf{I} \right) \right\} \quad (3-89)$$

where p denotes pressure, μ 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 coupled model formulations. The one-way coupled models sequentially solve for the fluid flow, compute the load from [Equation 3-89](#), 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 coupled versions offer computationally cheaper alternatives to a fully coupled counterpart.

Stationary and Time Dependent one-way coupled 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 coupled 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 coupled 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 micromechanics material properties cannot be directly used in a macroscopic scale in which the structure is to be analyzed. There the need to use a homogeneous material model with appropriate average 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 which represents the material microscopically.

The RVE typically is the smallest possible unit cell. 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 macro 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 the feature **Cell Periodicity**, 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, because 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. Macro scales are usually referred as homogenized continuous media and the micro scale refers to the heterogeneous cell/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

$$\bar{\sigma} = \frac{1}{V} \int_V \sigma dV \quad \text{and} \quad \bar{\varepsilon} = \frac{1}{V} \int_V \varepsilon dV$$

where V is the volume of the cell. The macroscopic elasticity matrix of the homogenized continuum is then defined from

$$\bar{\sigma} = \bar{D} : \bar{\varepsilon}$$

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.

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 , respectively. 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

$$\mathbf{u} = (R - I) \cdot (\mathbf{X} - \mathbf{X}_c) + \mathbf{u}_c$$

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 are fixed in space and do not follow the rotation of the rigid connector.

Contact Analysis Theory

In COMSOL Multiphysics you can model contact between a group of boundaries in 2D or 3D. There are two algorithms available: an *augmented Lagrangian* method and a *penalty* method.

In the augmented Lagrangian method, the system of equations is solved in a segregated way. Augmentation components are introduced for the contact pressure T_n and the components T_{t_i} of the friction traction vector \mathbf{T}_t . An additional iteration level is added where the usual displacement variables are solved separated from the contact pressure and traction variables. The algorithm repeats this procedure until it fulfills a convergence criterion.

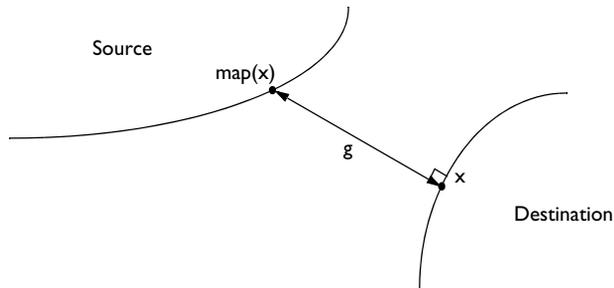
In the penalty method, no extra degrees of freedom are needed for the contact pressure or the friction traction vector. These results are just computed from the displacements and the penalty stiffness. This means that no special solver strategies are necessary.



- [Contact Modeling](#)
- Documentation of the [Contact](#), [Friction](#) and [Adhesion](#) features
- [Identity and Contact Pairs](#) in the *COMSOL Multiphysics Reference Manual*

In the following equations \mathbf{F} is the deformation gradient matrix. When looking at expressions evaluated on the destination boundaries, the expression $\text{map}(\mathbf{E})$ denotes the value of the expression \mathbf{E} evaluated at a corresponding source point, and g is the current gap distance between the destination and source boundary.

Both the contact map operator $\text{map}(\mathbf{E})$ and the gap distance variable are defined by the contact element `e1contact`. For each destination point where the operator or gap is evaluated, a corresponding source point is sought by searching in the direction normal to the destination boundary.



Before the boundaries come in contact, the source point found is not necessarily the point on the source boundary closest to the destination point. However, as the boundaries approach one another, the source point converges to the closest point as the gap distance approaches zero.

It is possible to add an offset value both to the source (o_{src}), and to the destination (o_{dst}). If an offset is used, the gap will be computed with the geometry treated as larger (or smaller, in the case of a negative offset) than what is actually modeled. The offset is given in a direction normal to the boundary. It can vary along the geometry, and may also vary in time (or by parameter value when using the continuation solver).

The effective gap distance d_g used in the analysis is thus

$$d_g = g - o_{\text{dst}} - \frac{o_{\text{src}}}{-(\mathbf{n} \cdot \text{map}(\mathbf{n}))}$$

The correction of the source side offset is necessary since the normal on the source side is not necessarily pointing toward the current point on the destination. When in contact, the normals will be exactly opposite.

In some cases, the discretization caused by the meshing of curved boundaries will give small irregularities in the gap values. You can then choose to adjust the gap by the computed gap value in the initial configuration, g_{init} . In this case, the effective gap distance is

$$d_g = g - g_{\text{init}} - o_{\text{dst}} - \frac{o_{\text{src}}}{-(\mathbf{n} \cdot \text{map}(\mathbf{n}))}$$

When including friction in the contact problem, it is important to keep track of not only the gap in the normal direction, but also the slip between the boundaries in the tangential direction. The slip \mathbf{s} since the last converged step is defined as

$$\mathbf{s} = \text{map}(F)(\mathbf{X}_m - \mathbf{X}_{m, \text{old}})$$

where the index ‘m’ indicates that the coordinates are taken as the mapped coordinates from the source side,

$$\mathbf{X}_m = \text{map}(\mathbf{X})$$

and $\mathbf{X}_{m, \text{old}}$ is the value of \mathbf{X}_m in the last converged time or parameter step. The coordinates are material (undeformed) coordinates. The slip vector is thus approximated using a backward Euler step. The deformation gradient F contains information about the local rotation and stretching.

AUGMENTED LAGRANGIAN METHOD

Using the special gap distance variable (`solid.gap`), the penalized contact pressure T_{np} is defined on the destination boundary as

$$T_{\text{np}} = \begin{cases} T_n - p_n d_g & \text{if } d_g \leq 0 \\ T_n e^{-\frac{p_n d_g}{T_n}} & \text{otherwise} \end{cases} \quad (3-90)$$

p_n is the user defined normal penalty factor.

The *penalized friction traction* \mathbf{T}_{tp} is defined on the destination boundary as:

$$\mathbf{T}_{tp} = \min\left(\frac{T_{t, \text{crit}}}{|\mathbf{T}_{t, \text{trial}}|}, 1\right)\mathbf{T}_{t, \text{trial}} \quad (3-91)$$

where $\mathbf{T}_{t, \text{trial}}$ is defined as

$$\mathbf{T}_{t, \text{trial}} = \mathbf{T}_t - p_t \mathbf{s} \quad (3-92)$$

In Equation 3-92, p_t is the user defined friction traction penalty factor. The critical sliding resistance, $T_{t, \text{crit}}$, is defined as

$$T_{t, \text{crit}} = \min(\mu T_{np} + T_{\text{cohe}}, T_{t, \text{max}}) \quad (3-93)$$

In Equation 3-93, μ is the friction coefficient, T_{cohe} is the user defined cohesion sliding resistance, and $T_{t, \text{max}}$ is the user defined maximum friction traction.

In the following equation δ is the variation (represented by the `test()` operator in COMSOL Multiphysics). The contact interaction gives the following contribution to the virtual work on the destination boundary:

$$\int_{\text{dst}} (T_{np} \delta g + \mathbf{T}_{tp} \cdot \text{map}(F) \delta \mathbf{X}_m) dA + \int_{\text{dst}} (w_{cn} \delta T_n + \mathbf{w}_{ct} \cdot \delta \mathbf{T}_t) dA$$

where \mathbf{w}_{cn} and \mathbf{w}_{ct} are contact help variables defined as:

$$\mathbf{w}_{cn} = \mathbf{T}_{np, i} - \mathbf{T}_{n, i+1}$$

$$\mathbf{w}_{ct} = (\gamma_{\text{fric}}(\mathbf{T}_{tp} - (\mathbf{n} \cdot \mathbf{T}_{tp})\mathbf{n}))_i - \mathbf{T}_{t, i+1}$$

where i is the augmented solver iteration number and γ_{fric} is a Boolean variable stating if the parts are in contact or not.

PENALTY METHOD

The penalty algorithm is essentially based on the insertion of a stiff spring between the contacting boundaries. The penalty factor can be interpreted as the stiffness of that spring. A high value of the stiffness fulfills the contact conditions more accurately, but if it is too high, it makes the problem ill-conditioned and unstable. The contact pressure in the normal direction is computed as

$$T_n = \begin{cases} -p_n d_g + p_0 & \text{if } d_g < p_0/p_n \\ 0 & \text{if } d_g \geq p_0/p_n \end{cases}$$

where dg is the effective gap distance, p_n is the penalty factor, and p_0 is the pressure at zero gap. The latter parameter 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 (a negative gap) in order for a contact pressure to develop.

In case of friction, the tangential force \mathbf{T}_t is computed as

$$\mathbf{T}_t = \min\left(\frac{T_{t, \text{crit}}}{|\mathbf{T}_{t, \text{trial}}|}, 1\right) \mathbf{T}_{t, \text{trial}}$$

where

$$\mathbf{T}_{t, \text{trial}} = -p_t \mathbf{s}$$

This can be viewed as a tangential spring giving a force proportional to the sliding distance. The penalty factor p_t can be identified as the spring constant. The sticking condition is thus replaced by a stiff spring, so that there is a small relative movement even if the force required for sliding is not exceeded.

The definition of $T_{t, \text{crit}}$ depends on the algorithm used for computing the normal contact.

$$T_{t, \text{crit}} = \begin{cases} \min(\mu T_{n_p} + T_{\text{cohe}}, T_{t, \text{max}}) & \text{augmented Lagrangian method} \\ \min(\mu T_n + T_{\text{cohe}}, T_{t, \text{max}}) & \text{penalty method} \end{cases}$$

The same notation as for the description of the augmented Lagrangian method above is used.

The springs between the contacting boundaries are added using weak expressions on the destination boundary.

$$\int_{\text{dst}} (T_{n_p} \delta g + \mathbf{T}_{t_p} \cdot \text{map}(F) \delta \mathbf{X}_m) dA$$

FRICITION MODELS

The friction model is either no friction or Coulomb friction.

The frictional coefficient μ is defined as

$$\mu = \begin{cases} \mu_d + (\mu_s - \mu_d) \exp(-\alpha_{\text{def}} |v_s|) & \text{if dynamic friction} \\ \mu_s & \text{otherwise} \end{cases}$$

where μ_s is the static frictional coefficient, μ_d is the dynamic frictional coefficient, v_s is the slip velocity, and α_{def} is a decay coefficient.

DIRECTIONS OF THE CONTACT FORCES

Since the contact model always makes use of the geometric nonlinearity assumption, the contact pressure and the friction force represent, respectively, the normal and tangential components of the nominal traction. This is the force density with components in the current configuration (that is, on the spatial frame) but related to the undeformed area of the corresponding surface element.

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 auxiliary degree of freedom located at Gauss points is used as an indicator of whether the adhesion criterion has been met or not.

When adhesion is active, the tensile stress σ_n in the normal direction is computed from the effective gap d_g as

$$\sigma_n = -K_p d_g$$

The default value for the stiffness in the normal direction K_p is the penalty stiffness of the contact condition, but you can also enter it explicitly. In compression, the penalty stiffness is always used. This leads to a bilinear stiffness in the normal direction when **Normal stiffness is User Defined**.

The shearing deformation is resisted using a shear stiffness K_τ , related to the normal stiffness by

$$K_\tau = n_\tau K_p$$

where n_τ is a coefficient with the default value 0.17. This coefficient can either be input explicitly, or be computed from a Poisson's ratio. A plane strain assumption is used for this conversion, giving

$$n_\tau = \frac{1 - 2\nu}{2(1 - \nu)}$$

The shear traction is proportional to the relative deformation between the source and destination boundaries measured from the position when the adhesion criterion was first fulfilled, so that

$$\mathbf{t} = -K_\tau \mathbf{s} F_t$$

where \mathbf{s} is the slip vector and F_t is the deformation gradient on the boundary.

DECOHESION

When adhesion is active, it is possible to break the bond between the source and destination boundaries by specifying a **Traction separation law**. Three different such decohesion laws are available.

The decohesion behavior in both tension or shear is based on two quantities, the maximum allowed stress and the energy release rate. Tearing in the normal direction is called mode I, and shear is called mode II. In the theory below, a subscript i can take the values I and II.

Up to the maximum stress, the adhesive layer is linearly elastic. At higher deformation, the stress decreases, and a damage is assumed to have occurred. The damage function is a scalar function of the maximum relative displacement which has occurred between the two boundaries. The stiffness in the damaged state is permanently reduced, even on unloading.

In general, the stress state is a combination of tension and shear, so the data for the two modes must be combined into a multiaxial decohesion rule. For two of the decohesion laws you can specify details of the multiaxial behavior.

As a measure of the displacement in the adhesive layer, the mixed mode relative displacement u_m , is defined as weighted combination of the normal direction displacement u_I and the tangential displacement u_{II} .

$$u_m = \sqrt{\langle u_I \rangle^2 + u_{II}^2}$$

In this expression, the notation

$$\langle x \rangle = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

has been used. Thus, u_m is insensitive to compressive displacement. It is u_m which is compared with the various mixed mode failure displacements described below.

Linear Separation Law

In the linear separation law, the decreasing part of the stress vs. displacement curve is linear. The graph in Figure 3-20 is generic, and valid both for mode I and mode II.

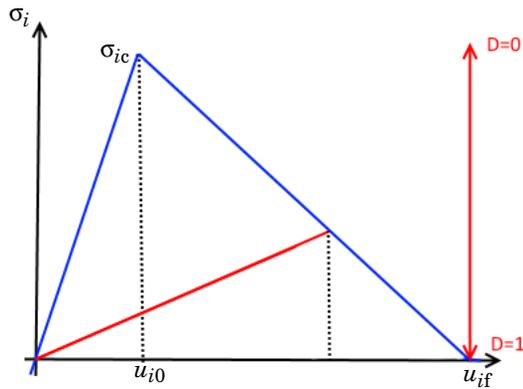


Figure 3-20: Stress vs. displacement curve for the linear separation law

The failure initiation displacement u_{i0} for each mode is determined from the maximum stress and the elastic stiffness, as

$$u_{I0} = \frac{\sigma_{Ic}}{K_p}$$

$$u_{II0} = \frac{\sigma_{IIc}}{K_\tau}$$

The decohesion displacement u_{if} for each mode is implicitly determined from the given energy release rate (which is the area under the stress vs. displacement curve):

$$u_{if} = \frac{2G_{ic}}{\sigma_{ic}}$$

The mixed mode failure initiation displacement is then defined as

$$u_0 = u_{I0}u_{II0} \sqrt{\frac{u_m^2}{\langle u_I \rangle^2 u_{II0}^2 + u_{II}^2 u_{I0}^2}}$$

The implication is that compressive displacements are ignored, and only tensile and tangential displacements contribute.

The final mixed mode failure displacement depends on the selected **Failure criterion**. The **Power Law** failure criterion is defined as

$$\left(\frac{G_I}{G_{Ic}}\right)^\eta + \left(\frac{G_{II}}{G_{IIc}}\right)^\eta = 1$$

whereas the **Benzeggagh-Kenane** failure criterion is defined as

$$G_{Ic} + (G_{IIc} - G_{Ic}) \left(\frac{G_{II}}{G_I + G_{II}}\right)^\eta = G_I + G_{II}$$

The exponent η is called the **Mode mixity exponent**.

From these relations, the total decohesion failure displacement in the mixed mode can be computed. For a power law criterion, the expression is

$$\left\{ \begin{array}{ll} u_f = \frac{2(1+\beta^2)}{u_0} \left(\left(\frac{K_p}{G_{Ic}}\right)^\eta + \left(\frac{\beta^2 K_\tau}{G_{IIc}}\right)^\eta \right)^{-\frac{1}{\eta}} & u_I > 0 \\ u_f = u_{II f} & u_I \leq 0 \\ \beta = \frac{u_{II}}{u_I} & \end{array} \right.$$

For the Benzeggagh-Kenane criterion, the corresponding expression is

$$\left\{ \begin{array}{ll} u_f = \frac{2u_m^2}{u_0 K_p (u_I^2 + n_\tau u_{II}^2)} \left(G_{Ic} + (G_{IIc} - G_{Ic}) \left(\frac{n_\tau u_{II}^2}{u_I^2 + n_\tau u_{II}^2} \right)^\eta \right) & u_I > 0 \\ u_f = u_{II f} & u_I \leq 0 \end{array} \right.$$

The damage function can be described as

$$D(u) = \min\left(1, \max\left(0, \frac{u_f(u - u_0)}{u_f - u_0}\right)\right)$$

The secant stiffness of the damaged adhesive layer (indicated by the red line in Figure 3-20) is

$$K_I = (1 - D)K_p$$

$$K_{II} = (1 - D)K_\tau$$

When $D > 0$, this stiffness is relevant only when the displacement is below the maximum experienced displacement, that is when moving on the red line in Figure 3-20, Figure 3-21, and Figure 3-22.

In an unloading state with $D > 0$, the stiffness is negative. The negative slope of the curve

Polynomial Separation Law

The polynomial separation law differs from the linear separation law in that the decreasing part of the stress curve is represented by a cubic polynomial. This polynomial is chosen so that it has zero slope in both ends, which means that the secant stiffness does not have any discontinuities. The area under the cubic function is the same as that of the linear function when the failure displacement u_{if} is the same. That is, for a given energy release rate G_{ic} , the same u_{if} is obtained.

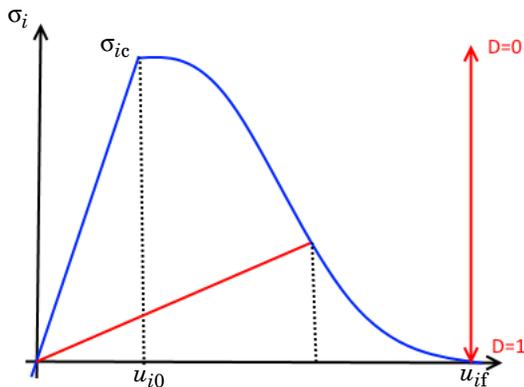


Figure 3-21: Stress vs. displacement curve for the polynomial separation law

All expressions presented under [Linear Separation Law](#) are still valid except the damage function, which now is

$$D(u) = \min\left(1, \max\left(0, \frac{u_0}{u} \left(1 + \left(\frac{u - u_0}{u_f - u_0}\right)^2 \left(2 \frac{u - u_0}{u_f - u_0} - 3\right)\right)\right)\right)$$

Multilinear Separation Law

The multilinear separation law introduces a region of constant stress, similar to a perfectly plastic material model, as shown in [Figure 3-22](#).

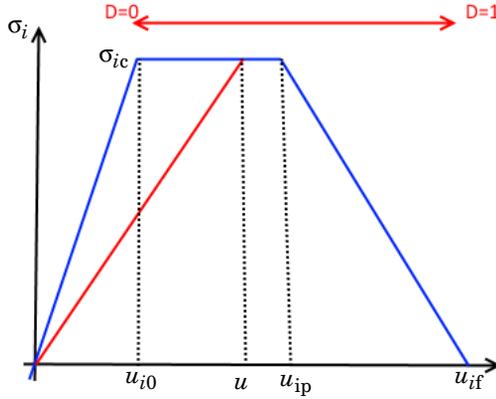


Figure 3-22: Stress vs. displacement curve for the multilinear separation law

This law requires one more material parameter λ , describing the width of the “plastic” region. The **Shape factor** λ is the ratio between the plastic (constant stress) part of G_{ic} and the total “inelastic” part of G_{ic} :

$$\lambda = \frac{\sigma_{ic}(u_{ip} - u_{i0})}{\sigma_{ic}(u_{ip} - u_{i0}) + \frac{1}{2}\sigma_{ic}(u_{if} - u_{ip})} = \frac{\sigma_{ic}(u_{ip} - u_{i0})}{G_{ic} - \frac{1}{2}\sigma_{ic}u_{i0}}$$

Note that the shape factor is assumed to be the same for both tension and shear. The value $\lambda = 0$ corresponds to the linear separation law.

The plastic displacement u_{ip} can thus be expressed as

$$u_{ip} = \lambda \frac{G_{ic}}{\sigma_{ic}} + u_{i0} \left(1 - \frac{\lambda}{2}\right)$$

The decohesion displacement is

$$u_{if} = (2 - \lambda) \frac{G_{ic}}{\sigma_{ic}} + \frac{\lambda u_{i0}}{2}$$

In this model, a linear mixed mode failure law is assumed:

$$\left(\frac{G_I}{G_{Ic}}\right) + \left(\frac{G_{II}}{G_{IIc}}\right) = 1$$

The mixed mode plastic failure displacement u_p is defined as

$$u_p = u_{I_p} u_{II_p} \sqrt{\frac{u_m^2}{\langle u_I \rangle^2 u_{II_p}^2 + u_{II}^2 u_{I_p}^2}}$$

The mixed mode failure displacement is

$$u_f = 2G_{Ic}G_{IIc} \frac{1 + \beta^2}{u_0(K_p G_{IIc} + \beta^2 K_\tau G_{Ic})} - (u_p - u_0)$$

The damage function can be written as

$$D(u) = \min \left(1, \max \left(0, \begin{cases} 1 - \frac{u_0}{u} & u < u_p \\ 1 - \frac{u_0}{u} \left(\frac{u_f - u}{u_f - u_p} \right) & u \geq u_p \end{cases} \right) \right)$$

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.

$$\sigma_{b(\max),ij} = \frac{6}{L} \int_0^L \sigma_{ij} \left(\frac{L}{2} - x_1\right) dx_1$$

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.

$$\begin{aligned}\sigma_{p(\text{start}), ij} &= [\sigma_{ij} - \sigma_{\text{mb}, ij}]_{x_1=0} = [\sigma_{ij}]_{x_1=0} - \sigma_{\text{m}, ij} + \sigma_{\text{b}(\text{max}), ij} \\ \sigma_{p(\text{end}), ij} &= [\sigma_{ij} - \sigma_{\text{mb}, ij}]_{x_1=L} = [\sigma_{ij}]_{x_1=L} - \sigma_{\text{m}, ij} - \sigma_{\text{b}(\text{max}), ij}\end{aligned}$$

The *stress intensity*, also known as the Tresca effective stress, is often the ultimate goal of a stress linearization. It is computed as a worst case of the effective stress at the two ends of the SCL. The stress intensity is computed from the principal stresses as

$$\sigma_{\text{int}} = \sigma_{\text{p1}} - \sigma_{\text{p3}}$$

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:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} \sigma_{\text{m}, 11} \\ \sigma_{\text{mb}, 22} \\ \sigma_{\text{mb}, 33} \\ \sigma_{\text{m}, 12} \\ \sigma_{\text{m}, 13} \\ \sigma_{\text{mb}, 23} \end{bmatrix}$$

Section forces, similar to what would be computed in a shell or plate analysis, is another type of result quantities available for each SCL. The in-plane forces are computed from the membrane stresses as

$$\begin{bmatrix} N_{22} \\ N_{33} \\ N_{23} \end{bmatrix} = L \begin{bmatrix} \sigma_{\text{m}, 22} \\ \sigma_{\text{m}, 33} \\ \sigma_{\text{m}, 23} \end{bmatrix}$$

The bending moments are computed from the bending stresses as

$$\begin{bmatrix} M_{22} \\ M_{33} \\ M_{23} \end{bmatrix} = \frac{L^2}{6} \begin{bmatrix} \sigma_{\text{b}(\text{max}), 22} \\ \sigma_{\text{b}(\text{max}), 33} \\ \sigma_{\text{b}(\text{max}), 23} \end{bmatrix}$$

The out-of-plane shear forces are computed from the membrane stresses as

$$\begin{bmatrix} Q_2 \\ Q_3 \end{bmatrix} = L \begin{bmatrix} \sigma_{m,12} \\ \sigma_{m,13} \end{bmatrix}$$

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

$$W_s = \int_0^{\varepsilon_{el}} \sigma : d\varepsilon_{el} \quad (3-94)$$

If the linear elasticity is assumed, then

$$\sigma = \sigma_i + C : \varepsilon_{el}$$

where σ_i is the initial stress. The integration can then be carried out analytically and the result is

$$W_s = \sigma_i : \varepsilon_{el} + \frac{1}{2} (C : \varepsilon_{el}) : \varepsilon_{el} = \frac{1}{2} (\sigma_i + \sigma) : \varepsilon_{el}$$

This expression is used for the Linear Elastic Material model in a Stationary or Time Dependent analysis. An implication is that if you modify the linear elastic model in a way that violates the assumption about stress-strain linearity above, then the computed strain energy density may be wrong, for example, using a strain dependent Young's modulus or a nonconstant initial stress.

In the case of Frequency Domain analysis, only the harmonic part is considered. That is, a constant prestress does not contribute to the strain energy density. To emphasize this, the concept of *stored energy* is used. The stored energy is the cycle average of the elastic energy; that is,

$$W_h = \frac{1}{T} \int_0^T \frac{1}{2} \boldsymbol{\sigma}(t) : \boldsymbol{\varepsilon}(t) dt$$

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

$$W_h = \frac{1}{4} \text{real}(\boldsymbol{\sigma} : \bar{\boldsymbol{\varepsilon}})$$

where the stress and strain are considered as complex quantities, and the overbar 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 3-94](#) 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 addition of the Multibody Dynamics Module.

Kinetic Energy

For all materials, a kinetic energy density is computed. For a Time Dependent study, it is defined as

$$W_k = \frac{1}{2}\rho v^2$$

where ρ is the mass density, and v is the velocity.

In Frequency Domain, the 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

$$\mathbf{I} = -\sigma \cdot \mathbf{v}$$

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:

$$\mathbf{I} = -\sigma \cdot \tilde{\mathbf{v}}$$

The *mechanical energy flux* is in the frequency domain defined as a real quantity, the cycle average of the complex mechanical energy flux.

$$\mathbf{I} = \frac{1}{2} \text{real}(-\sigma \cdot \tilde{\mathbf{v}})$$

Energy Variables

The energy variables used in the Structural Mechanics interfaces are summarized in [Table 3-6](#).

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE	COMMENT
phys.Ws	Elastic strain energy density	J/m ³	solid, mbd, shell, plate, mbrn, beam, truss	
phys.WsM	Membrane strain energy density	J/m ²	shell, plate	Contributes to phys.Ws
phys.WsB	Bending strain energy	J/m ²	shell, plate	Contributes to phys.Ws
phys.WsS	Shear strain energy	J/m ²	shell, plate	Contributes to phys.Ws
phys.WsM	Membrane strain energy 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 ³	solid, mbd, shell, plate, mbrn, beam, truss	

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE	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 ³	solid	Energy density dissipated by plasticity
phys.Wp_tot	Total plastic dissipation	J	solid	Global variable containing integration and summation of all phys.Wp contributions in a physics interface.
phys.Wc	Creep dissipation density	J/m ³	solid	Energy density dissipated by creep.
phys.Wc_tot	Total creep dissipation	J	solid	Global variable containing integration and summation of all phys.Wp contributions in a physics interface.
phys.Wv	Viscoelastic dissipation density	J/m ³	solid	Energy density dissipated by viscosity.
phys.Wv_tot	Total viscoelastic dissipation	J	solid	Global variable containing integration and summation of all phys.Wv contributions in a physics interface.
phys.Wed	Dielectric dissipation density	J/m ³	solid	Piezoelectric electrical damping

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE ¹	COMMENT
phys.Wed_tot	Total dielectric dissipation	J	solid	Global variable containing integration and summation of all phys.Wed contributions in a physics interface.
phys.Wmd	Damping dissipation density	J/m ³	solid	Piezoelectric mechanical damping
phys.Wmd_tot	Total damping dissipation	J	solid	Global variable containing integration and summation of all phys.Wmd contributions in a physics interface.
phys.l*	Mechanical energy flux, *coordinate	W/m ²	solid, mbd	
phys.lcomplex*	Complex mechanical energy flux, *coordinate	W/m ²	solid, mbd	Frequency domain only

¹The 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. Souza et al. "Three-dimensional model for solids undergoing stress-induced phase transformations," *European J. Mech. A Solids*, vol. 17, pp. 789–806, 1988.
19. 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.
20. D. Lagoudas (Ed.) *Shape Memory Alloys: Modeling and Engineering Applications*, Springer, 2008.
21. L. Mullins, "Effect of stretching on the properties of rubber," *Journal of Rubber Research*, vol. 16, pp. 275–289, 1947.
22. L. Mullins and N. R. Tobin, "Theoretical model for the elastic behavior of filled-reinforced vulcanized rubbers," *Rubber Chem. Technol.* 30, pp. 555–571, 1957.
23. L. Mullins. "Softening of rubber by deformation," *Rubber Chem. Technol.* 42, pp. 339–362, 1969.
24. R. Ogden and D. Roxburgh, "A pseudo-elastic model for the Mullins effect in filled rubber," *Proc. R. Soc. Lond. A* 455, pp. 2861–2878, 1999.
25. A. Dorfmann and R. Ogden "A constitutive model for the Mullins effect with permanent set in particle-reinforced rubber," *Int J Solids Struct* 41, pp. 1855-1878, 2004.
26. 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* 25, pp. 877–896, 2006.
27. C. Miehe. "Discontinuous and continuous damage evolution in Ogden-type large strain elastic materials," *European Journal of Mechanics, A/Solids* 14, pp. 697-720, 1995.

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. B. Bresler and K.S. Pister, “Strength of Concrete Under Combined Stresses,” *ACI Journal*, vol. 551, no. 9, pp. 321–345, 1958.
3. H. Matsuoka and T. Nakai, “Stress-deformation and Strength Characteristics of Soil Under Three Different Principal Stresses,” *Proc. JSCE*, vol. 232, 1974.
4. 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.
5. H.S. Yu, *Plasticity and Geotechnics*, Springer, 2006.
6. V. Marinou, P. Marinou, and E. Hoek, “The Geological Strength Index: Applications and Limitations,” *Bull. Eng. Geol. Environ.*, vol. 64, pp. 55–65, 2005.
7. J. Jaeger, N. G. Cook, and R. Zimmerman, *Fundamentals of Rock Mechanics*, 4th ed., Wiley-Blackwell, 2007.
8. G. C. Nayak and O. C. Zienkiewicz, “Convenient Form of Stress Invariants for Plasticity,” *J. Struct. Div. ASCE*, vol. 98, pp. 949–954, 1972.
9. 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.
10. 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.
11. B.H.G. Brady and E.T. Brown, *Rock Mechanics for Underground Mining*, 3rd ed., Springer, 2004.
12. H.A. Taiebat and J.P. Carter, “Flow Rule Effects in the Tresca Model,” *Computer and Geotechnics*, vol. 35, pp. 500–503, 2008.
13. 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.

14. D.M. Wood, *Soil Behaviour and Critical State Soil Mechanics*, Cambridge University Press, 2007.
15. D.M. Potts and L. Zdravkovic, *Finite Element Analysis in Geotechnical Engineering*, Thomas Telford, 1999.
16. W. Tiejcheng and others, *Stress-strain Relation for Concrete Under Triaxial Loading*, 16th ASCE Engineering Mechanics Conference, 2003.
17. W.F. Chen, *Plasticity in Reinforced Concrete*, McGraw-Hill, 1982.
18. N. Ottosen, "A Failure Criterion for Concrete," *J. Eng. Mech. Division, ASCE*, vol. 103, no. 4, pp. 527–535, 1977.
19. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modelling*, Elsevier, 2005.
20. 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.
21. M. Lui, and J.P. Carter, *A structured Cam Clay model*, *Can. Geotech. J.*, vol. 39, pp. 1313-1332, 2002.
22. E. Alonso, A. Gens and A. Josa, *A constitutive model for partially saturated soils*, *Geotechniques*, vol.40, 1990.
23. 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.
24. T. Bower, *Constitutive modelling of soils and fibre-reinforced soils*, PhD Thesis, Cardiff University 2017.

REFERENCES TO THE NONLINEAR ELASTIC MATERIAL THEORY

1. Ramberg, W. and Osgood, W. R., "Description of stress-strain curves by three parameters," *NACA Technical Note*, no. 902, 1943.
2. Bower, A. F., *Applied Mechanics of Solids*, CRC Press, 2009.
3. Hardin, B.O., and Drnevich, V.P., "Shear modulus and damping in soils: Design equations and curves," *J. Soil Mechanics and Foundations Division, ASCE* 98, pp. 667–692, 1972.

4. Duncan, J.M. and Chang, C.Y., “Nonlinear analysis of stress and strain in soils,” *J. Soil Mechanics and Foundations Division*, vol. 96, no. 5, pp. 1629–1653, 1970.
5. Duncan, J.M., Byrne, P., Wong, K.S., and Mabry, P., “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. Selig, E.T., *Soil Parameters for Design of Buried Pipelines, Pipeline Infrastructure*, B. A. Bennett, ed., ASCE, New York, pp. 99–116, 1988.

REFERENCES FOR ELASTOPLASTIC MATERIALS

7. 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.
8. M. Jirasek and Z. Bazant, *Inelastic Analysis of Structures, Ch.20 General Elastoplastic Constitutive Models*. Wiley, 2001.
9. J. Simo and T. Hughes, *Computational Inelasticity*, Springer, 1998.
10. 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.
11. J. Lubliner, *Plasticity Theory*, Dover, 2008.
12. R. Hill, “A Theory of the Yielding and Plastic Flow of Anisotropic Metals,” *Proc. Roy. Soc. London*, vol. 193, pp. 281–297, 1948.
13. N. Ottosen and M. Ristinmaa, *The Mechanics of Constitutive Modeling*, Elsevier Science, 2005.
14. S. Shima and M. Oyane. “Plasticity theory for porous metals,” *International Journal of Mechanical Sciences*, vol. 18, pp. 285–291, 1976.
15. 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.
16. V. Tvergaard and A. Needleman, “Analysis of the cup-cone fracture in a round tensile bar,” *Acta Metallurgica*, vol. 32, pp. 157–169, 1984.
17. 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.

18. 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.
19. 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.
20. P. Redanz, “Numerical modelling of the powder compaction of a cup,” *European Journal of Mechanics - A/Solids*, vol. 18, pp.399– 413, 1999.
21. J. Cedergren, N. Sorensen, and A. Bergmark, “Three-dimensional analysis of compaction of metal powder,” *Mechanics of Materials*, vol. 34, pp. 43–59, 2004.
22. S. Tsai and E. Wu, “A general theory of strength for anisotropic materials,” *Journal of Composite Materials*, vol. 5, pp. 58-80, 1971.
23. P. Hopkins, “Benchmarks for Membrane and Bending Analysis of Laminated Shells. Part 2: Strength Analysis,” NAFEMS Ltd, 2005.

REFERENCES FOR THE DAMAGE MODELS

1. R. de Borst, M. Crisfield, J. Remmers, 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, B. Oh, “Crack band theory for fracture in concrete,” *Materials and Structures*, vol. 16, pp. 155-177, 1983.
5. M. Jirásek, M. Bauer, “Numerical aspects of the crack band approach,” *Computers and Structures*, vol. 110, pp. 60-78, 2012.
6. R. Peerlings et. al., “Gradient enhanced damage for quasi-brittle materials,” *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 3391-3403, 1996.

REFERENCES FOR THE PIEZOELECTRIC DEVICES 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. 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,” *The 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. De Lorenzis L., Fernando D., Teng J.G. “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, 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, Issue 5, 1 March 2008, pp. 1497–1512.
3. S T Pinho, L Iannucci, P Robinson, “Formulation and implementation of decohesion elements in an explicit finite code,” *Composites Part A: Applied Science and Manufacturing*, vol. 37, Issue 5, May 2006, pp. 778–789.

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 or 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 <http://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** (the default), **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 zero stress criterion.

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.

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** . Note that 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** (the default) or **Quasi-static**. Use **Quasi-static** 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 **Quasi-static** is selected.

For problems with creep, and sometimes viscoelasticity, the problem can be considered as quasi-static. 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 coordinates for the **Reference point for moment computation** \mathbf{x}_{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

The typical wave speed c_{ref} is a parameter for the perfectly matched layers (PMLs) if used in a solid wave propagation model. The default value is `solid.cp`, the pressure-wave speed. To use another wave speed, enter a value or expression in the **Typical wave speed for perfectly matched layers** field.

Note: This section is only available with COMSOL products that include PMLs (see <http://www.comsol.com/products/specifications/>)

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 2-3](#) for links to common sections and [Table 2-4](#) 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
- Edge Load
- 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)
- 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
- Stress Linearization
- Symmetry
- Thin Elastic Layer
- Thin-Film Damping
- Thread Boundary Selection



In the *COMSOL Multiphysics Reference Manual*, see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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
- Dielectric Loss
- External Stress
- Fixed Constraint (Rigid Domain)
- Friction
- Hygroscopic Swelling
- Initial Stress and Strain
- Initial Values (Rigid Domain)
- 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
- Soil Plasticity
- 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

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.

LAYER SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the [Initial Values](#) node.

INITIAL VALUES

Enter values or expressions for the initial values of the **Displacement field \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 available in:

- Domains in 2D for the Solid Mechanics and Multibody Dynamics interfaces.
- Boundaries in 3D for the Membrane and Shell interfaces. (See also [Thickness and Offset](#) in shell interface documentation).
- Boundaries in 2D axisymmetry for the Membrane interface.

CHANGE THICKNESS

Enter a value for the **Thickness** d . This value replaces the overall thickness for the selected domains or boundaries.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Change Thickness

Multibody Dynamics>Change Thickness

Membrane>Change Thickness

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Domains>Solid Mechanics>Change Thickness

Domains>Multibody Dynamics>Change Thickness

Physics tab with **Membrane** selected:

Boundaries>Membrane>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">• Thermal Expansion (for Materials)• Hygroscopic Swelling• Initial Stress and Strain• External Stress• External Strain• Damping• Viscoelasticity• Plasticity• Creep	<ul style="list-style-type: none">• Viscoplasticity• Porous Plasticity• Soil Plasticity• Concrete• Rocks• Damage• Activation• Safety
---	---

Note: Some options are only available with certain COMSOL products (see <http://www.comsol.com/products/specifications/>). Also, the available options depend on the physics interface in which the **Linear Elastic Material** is used.

LAYER SELECTION

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.

LINEAR ELASTIC MATERIAL

Define the **Solid model** and the linear elastic material properties.

Solid Model

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 , ν_{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 <http://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** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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	$\nu =$	ν	$\frac{E}{2G} - 1$	$\frac{1}{2} \left(1 - \frac{3G}{3K+G} \right)$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	$K =$	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{3(3G-E)}$	K	$\lambda + \frac{2\mu}{3}$
Shear modulus	$G =$	$\frac{E}{2(1+\nu)}$	G	G	μ
Lamé parameter λ	$\lambda =$	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{G(E-2G)}{3G-E}$	$K - \frac{2G}{3}$	λ

TABLE 4-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Lamé parameter μ	$\mu =$	$\frac{E}{2(1+\nu)}$	G	G	μ
Pressure-wave speed	$c_p =$	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\frac{K+4G/3}{\rho}}$	$\sqrt{\frac{\lambda+2\mu}{\rho}}$
Shear-wave speed	$c_s =$	$\sqrt{\frac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

The individual property parameters are:

- **Young's modulus** (elastic modulus) E .
- **Poisson's ratio** ν .
- **Shear modulus** G .
- **Bulk modulus** K .
- **Lamé parameter** λ and **Lamé parameter** μ .
- **Pressure-wave speed** (longitudinal wave speed) c_p .
- **Shear-wave speed** (transverse wave speed) c_s . This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Specification of Elastic Properties for Orthotropic Materials

When **Orthotropic** is selected from the **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** ν , 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

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$



ν_{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 <http://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 deformation 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** button () and select **Advanced Physics Options**.

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">• Thermal Expansion (for Materials)• Hygroscopic Swelling• Initial Stress and Strain• External Stress• External Strain• Damping• Viscoelasticity• Plasticity	<ul style="list-style-type: none">• Creep• Viscoplasticity• Porous Plasticity• Soil Plasticity• Concrete• Rocks• Safety
---	---

Note: Some options are only available with certain COMSOL products (see <http://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, 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 ρ** uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input**

section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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** ν .
- **Shear modulus** G .
- **Bulk modulus** K .
- For **Ramberg-Osgood**:
 - **Reference stress** σ_{ref} .
 - **Reference strain** ε_{ref} .
 - **Stress exponent** n .
- For **Power law** and **Hyperbolic law**:
 - **Reference shear strain** γ_{ref} .
 - **Strain exponent** n .
- For **Hardin-Drnevich**, define the **Reference shear strain** γ_{ref} .
- For **Duncan-Chang**, define the **Ultimate deviatoric stress** q_{ult} .
- For **Duncan-Selig**:
 - **Ultimate deviatoric stress** q_{ult} .
 - **Ultimate strain** ε_{ult} .

Uniaxial Data

For **Uniaxial data** enter a value or expression for the **Uniaxial stress function** σ_{ax} as a function of the uniaxial strain. The default expression is the linear function $(210[\text{GPa}]) * \text{solid.eax N/m}^2$, which corresponds to a linear elastic material with Young's modulus 210 GPa.

From the **Specify** list select how to specify the second elastic property for the material — **Bulk modulus** or **Poisson's ratio**. Then, depending on the selection, enter a value or select from the applicable list to use the value **From material** or enter a **User defined** value or expression:

- **Bulk modulus** K .
- **Poisson's ratio** ν .

When you select **Bulk modulus**, the Young's modulus is computed from the tensile part of the **Uniaxial stress function** σ_{ax} . When you select **Poisson's ratio**, you can either use the tensile part (default), or use the full tensile-compressive function by selecting the check box **Use nonsymmetric stress-strain data**.

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** (the default) or enter a **User defined** value or expression.

- **Pressure** p . The default expression is `(-160[GPa])*solid.eelvo1` N/m², which corresponds to a linear elastic material with bulk modulus 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 deformation 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** button () and select **Advanced Physics Options**.

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 <http://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 a **External Stress** feature 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** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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** ν or the **Shear modulus** G , **Density** ρ , **Slope of critical state line** M , **Swelling index** κ , **Compression index** λ , and **Void ratio at reference pressure** e_{ref} are taken **From material**. For **User defined** enter other values or expressions

Enter a value or expression for the **Reference pressure** p_{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** ϕ as **From material** (the default) or **User defined**.



Isotropic Compression with the 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** ν or the **Shear modulus** G , **Density** ρ , **Slope of critical state line** M , **Swelling index for structured clay** κ_s , **Compression index for destructured clay** λ_d , **Angle of internal friction** ϕ , **Initial structure strength** p_{bi} , **Destructuring index for volumetric deformation** d_v , **Destructuring index for shear deformation** d_s , **Plastic potential shape parameter** ξ , **Void ratio at reference pressure for destructured clay** $e_{\text{ref}d}$, **Additional void ratio at initial yielding** Δe_i , and **Critical effective 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_{ref} , and the **Initial consolidation pressure** p_{c0} .



See also [The Modified Structured Cam-Clay Soil Model](#) in the *Structural Mechanics Theory* chapter.

Extended Barcelona Basic

The defaults for the **Shear modulus** G , **Density** ρ , **Swelling index for changes in suction** κ_s , **Compression index at saturation** λ_0 , **Angle of internal friction** ϕ , **Weight parameter** w , **Soil stiffness parameter** m , **Plastic potential shape parameter** b_s , **Tension to suction ratio** k ,

Void ratio at reference pressure and saturation $e_{\text{ref}0}$, 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 , the **Reference pressure** p_{ref} , and the **Initial consolidation pressure** p_{c0} .



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}^{ref} , **Reference stiffness for unloading and reloading** $E_{\text{ur}}^{\text{ref}}$, **Elastoplastic compression modulus** K_c , **Poisson's ratio** ν , **Density** ρ , **Stress exponent** m , **Cohesion** c , **Angle of internal friction** ϕ , **Dilatation angle** ψ , and **Void ratio at reference pressure** e_{ref} 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_{ref} , and the **Initial consolidation pressure** p_{c0} .

Select the **Include dilatancy cut-off** check box if needed. The defaults for the **Maximum void ratio** e_{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 deformation 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** button () and select **Advanced Physics Options**.

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 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">• Thermal Expansion (for Materials)• Hygroscopic Swelling• External Stress• External Strain | <ul style="list-style-type: none">• Damping• Viscoelasticity• Mullins Effect• Plasticity |
|--|---|



See also [Hyperelastic Material Models](#) in the *Structural Mechanics Theory* 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** ρ uses values **From material**. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

- [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](#)
- [User defined](#)

Neo-Hookean

The default values for both **Lamé parameter λ** and **Lamé parameter μ** use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus κ** instead of the Lamé parameter λ to define the volumetric strain energy density. If the **Incompressible material** option is selected from the **Compressibility** list, enter the Lamé parameter μ only.

St Venant-Kirchhoff

The default values for both **Lamé parameter λ** and **Lamé parameter μ** use values **From material**. If the **Incompressible material** option is selected from the **Compressibility** list, enter only the Lamé parameter μ .

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 κ** .

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 κ** .

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 κ** .

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 κ** .

Ogden

In the table for the **Ogden parameters**, enter values or expressions in each column: **p** , **Shear modulus (Pa)**, and **Alpha parameter**.

If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus κ** .

Storakers

For **Storakers**, in the table for the **Storakers parameters**, enter values or expressions in each column: **p** , **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** κ .

Arruda-Boyce

For **Arruda-Boyce** the default values for the **Macroscopic shear modulus** μ_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** κ .

Gent

For **Gent** the default values for the **Macroscopic shear modulus** μ and the model parameter j_m is to use values **From material**. If the **Nearly incompressible material** option is selected from the **Compressibility** list, enter the **Bulk modulus** κ .

Blatz-Ko

For **Blatz-Ko** the **Shear modulus** μ and the **Model parameters** β and ϕ all use values **From material**.

Gao

For **Gao** the **Model parameters** a and n all use values **From material**.

Murnaghan

For **Murnaghan** the **Murnaghan third-order elastic moduli** constants l , m , and n and the **Lamé parameters** λ and μ use values **From material**.

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_{iso} and the **Volumetric strain energy density** W_{vol} .

If **Incompressible material** is selected, enter the **Isochoric strain energy density** W_{iso} 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** button () and select **Advanced Physics Options**.

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
Membrane>Material Models>Hyperelastic Material**

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>Material Models>Hyperelastic Material

Physics tab with **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. It is available in the Solid Mechanics interface. 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 <http://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. 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.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



See also

- [Mass Density and Volume Reference Temperature](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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** enter the **Shape memory alloy reference temperature** T_0 . The defaults for the **Poisson's ratio** ν and **Density** ρ , are taken **From material**. For **User defined** enter other values or expressions.

For **Austenite**, select a material from the list. The **Young's modulus** E_A 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**, 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 , the **Slope of austenite limit curve** C_A , and the **Maximum transformation strain** $\epsilon_{tr,max}$.

Under **Phase transformation kinetics**, select the **Transformation function** from the list: **Quadratic**, **Cosine**, **Smooth** or **User defined**.

For **Smooth**, enter the smoothing parameters η_1 , η_2 , η_3 , and η_4 .

For **User defined**, enter the **Yield stress** σ_{ys} , the **Forward transformation law**, and the **Reverse transformation law**.

Under **Phase transformation direction**, select the **Transformation direction** from the list: **Automatic** or **User defined**.

Souza-Auricchio

For **Souza-Auricchio** the defaults for the **Poisson's ratio** ν and **Density** ρ , are taken **From material**. For **User defined** enter other values or expressions.

For **Austenite**, select a material from the list. The **Young's modulus** E_A is taken from the selected material. For **Martensite**, select a material from the list. The **Young's modulus** E_M is taken from the selected material. For **User defined** enter other values or expressions.

Under **Phase transformation parameters**, enter the **Martensite finish temperature** M_f , the **Slope of limit curve** β , the **Maximum transformation strain** $\epsilon_{tr,max}$, the **Initial yield stress** σ_{ys0} , the **Hardening modulus** H_k , and the **Indicator function coefficient** γ .

Initial Transformation State

For Laogudas model, enter the **Initial martensite volume fraction**, the **Initial transformation strain tensor** and the **Initial transformation strain tensor at reverse point**.

For 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** button () and select **Advanced Physics Options**.

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

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>Material Models>Shape Memory Alloy

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 **Piezoelectric Devices** 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 structural mechanics interface in the absence of an active **Piezoelectric Effect** multiphysics coupling node the material behaves similarly to a [Linear Elastic Material](#) node, with elastic properties as corresponding to the elasticity or compliance matrix entered (see below). The piezoelectric effect is then *not* included in the corresponding 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 (ordering: xx, yy, zz, yz, xz, xy)** (c_E).
- For a **Strain-charge form**, select a **Compliance matrix (ordering: xx, yy, zz, yz, xz, xy)** (s_E).
- Select a **Coupling matrix (ordering: xx, yy, zz, yz, xz, xy)** (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.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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 (ordering: xx, yy, zz, yz, xz, xy)** (s_H).
- For a **Stress-magnetization form**, select an **Elasticity matrix (ordering: xx, yy, zz, yz, xz, xy)** (c_H).
- Select a **Piezomagnetic coupling matrix (ordering: xx, yy, zz, yz, xz, xy)** (d_{HT} or e_{HS}).
- Select a **Relative permeability** (μ_{RT} or μ_{RS}).
- Select a **Density** (p).



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.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
 - [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.
-

NONLINEAR MAGNETOSTRICTIVE MATERIAL PROPERTIES

See the corresponding the 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, Membrane, and

Layered Shell interfaces, and can be used together with [Linear Elastic Material](#), [Nonlinear Elastic Material](#), and [Hyperelastic Material](#).



See also [Linear Viscoelastic Materials](#) and [Large Strain Viscoelasticity](#) in the *Structural Mechanics Theory* chapter.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Viscoelasticity](#) node.

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** (the default), **Williams-Landel-Ferry**, **Arrhenius**, 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 then the shift function $a_T(T)$ is computed from these parameters and the relaxations time is shifted according to it:
 - **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 then the shift function $a_T(T)$ is computed from these parameters and the relaxations time is shifted according to it:
 - **Reference temperature** T_0 . The default is 293.15 K.
 - **Activation energy** Q .
- For **User defined** enter a value or expression for the shift function a_T .

VISCOELASTICITY MODEL

Select a **Material model** — **Generalized Maxwell** (the default), **Standard linear solid**, or **Kelvin-Voigt**. Then see the settings for each option that follows.

For any material model, you can select the shear modulus to use when solving a stationary problem. Choose the **Static stiffness** for the material model — **Long-term** (the default) or **Instantaneous**.

Generalized Maxwell

For **Generalized Maxwell** in the table 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 τ_m in the **Relaxation time (s)** column for the spring-dashpot pair in branch m .

For large strain viscoelasticity, in each **Branch** row enter β_m (the energy factor of the branch) in the **Energy factor (I)** column and the relaxation time constant τ_m in the **Relaxation time (s)** column for the spring-dashpot pair.

- Use the **Add** button () to add a row to the table and the **Delete** button () to delete a row in the table.
- Using the **Load from file** button () and the **Save to file** button () 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.

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** τ_v . The default is 3000 s.
- **Viscosity** η_v of the dash-pot. 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.

For large strain viscoelasticity, enter the **Relaxation time** τ_v , which default is 3000 s, and the **Energy factor** β_v of the dash-pot. The default is 0.2.

Kelvin-Voigt

For **Kelvin-Voigt** enter the values for the parameter that describes the viscous behavior of the single dash-pot.

For linear viscoelasticity, select an option from the **Relaxation data** list and edit the default as needed:

- **Relaxation time** τ_v . The default is 3000 s.
- **Viscosity** η_v of the dash-pot. The default is $6 \cdot 10^{13}$ Pa·s.

For large strain viscoelasticity, enter the **Relaxation time** τ_v . The default is 3000 s.

Burgers

For **Burgers** enter the values for the parameter that describes the viscous behavior of the spring dash-pot in series with a second spring-dash-pot 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 dash-pots τ_{v1} and τ_{v2} .
- **Viscosity**. Enter the viscosity of the dash-pots. The default is $6 \cdot 10^{13}$ Pa·s for both η_{v1} and η_{v2} .

In the **Shear modulus** field, enter the stiffness of the second spring G_{v2} . The default is $2 \cdot 10^{10}$ Pa.

DISCRETIZATION

Select a **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**
- Layered Shell>Linear Elastic Material>Viscoelasticity**
- Membrane>Linear Elastic Material>Viscoelasticity**
- Membrane>Nonlinear Elastic Material>Viscoelasticity**
- Membrane>Hyperelastic Material>Viscoelasticity**

Ribbon

Physics tab with **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](#).

Damage function

The **Damage function** defines the model for Mullins effect.

For **Ogden-Roxburgh** enter the following settings:

- **Maximum damage** d_{∞} . The default is 1.
- **Damage saturation** W_{sat} . The default is 1 MJ/m³.
- **Deformation dependence coefficient** α . The default is 0.

For **Miehe** enter the following settings:

- **Maximum damage** d_{∞} . The default is 1.

- **Damage saturation** W_{sat} . The default is 1 MJ/m³.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Hyperelastic Material>Mullins Effect

Membrane>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 is available in the Solid Mechanics, 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 <http://www.comsol.com/products/specifications/>.



See also [Elastoplastic Material Models](#) in the *Structural Mechanics Theory* 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 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 non associated 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** σ_{ys0ij} or **Hill’s coefficients** $F, G, H, L, M,$ and N . The default for either selection uses values **From material** (if it exists) or **User defined**. The principal directions of orthotropy are inherited from the coordinate system selection in the **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** (the default), **von Mises**, or **User defined** (non associated). Enter a **User defined** value in the Q field as needed.

Initial Yield Stress

For all yield criteria, the default **Initial yield stress** σ_{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 σ_{ys} is modified as hardening occurs, and it is related to the effective plastic strain ϵ_{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 effective plastic strain ϵ_{pe} . The Young’s modulus E is taken from the elastic material properties.

- Select **Ludwik** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the power-law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + k\epsilon_{pe}^n$$

the **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.

- For **Swift** nonlinear isotropic hardening, the **Reference strain** ϵ_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified by the power-law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} \left(1 + \frac{\epsilon_{pe}}{\epsilon_0} \right)^n$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + \sigma_{sat} \left(1 - e^{-\beta \epsilon_{pe}} \right)$$

the **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.

- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_{∞} , the **Saturation coefficient** m , and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{ys}(\epsilon_{pe}) = \sigma_{ys0} + (\sigma_{\infty} - \sigma_{ys0}) \left(1 - e^{-m \epsilon_{pe}^n} \right)$$

- For **User defined** isotropic hardening the **Hardening function** $\sigma_h(\epsilon_{pe})$ uses values **From material**. The yield level σ_{ys} is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\epsilon_{pe})$$

This definition implies that the hardening function $\sigma_h(\epsilon_{pe})$ 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 fit nonlinear isotropic hardening curves. The hardening function can depend on more variables than the effective plastic strain, for example the temperature.

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 σ_b as plasticity occurs:

$$\sigma_b = C_k \cdot \frac{2}{3} \varepsilon_p \quad \text{with} \quad \frac{1}{C_k} = \frac{1}{E_k} - \frac{1}{E}$$

This is Prager's linear kinematic hardening model, so the back stress σ_b is collinear to the plastic strain tensor ε_p .

- If **Armstrong-Frederick** is selected from the list, the default **Kinematic hardening modulus** C_k and **Kinematic hardening parameter** γ_k use values **From material**. These parameters are used to calculate the back stress σ_b from the rate equation

$$\dot{\sigma}_b = \frac{2}{3} C_k \dot{\varepsilon}_p - \gamma_k \dot{\varepsilon}_p \sigma_b$$

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:

$$\dot{\sigma}_{b,i} = \frac{2}{3} C_i \dot{\varepsilon}_p - \gamma_i \dot{\varepsilon}_p \sigma_{b,i}$$

The total back stress σ_b is then computed from the sum

$$\sigma_b = \frac{2}{3} C_0 \varepsilon_p + \sum_{i=1}^N \sigma_{b,i}$$

For each **Branch** row, enter C_i (the hardening modulus of the branch i) in the **Hardening modulus (Pa)** column and γ_i (the hardening parameter of the branch i) in the **Hardening parameter (1)** column.

Use the **Add** button () and the **Delete** button () to add or delete a row in the table. Use the **Load from file** button () and the **Save to file** button () to load

and store data for the branches in a text file with three space-separated columns (from left to right): the branch number, the hardening modulus for that branch, and the hardening parameter for that branch.



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**.
-

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Linear Elastic Material>Plasticity

Solid Mechanics>Nonlinear Elastic Material>Plasticity

Solid Mechanics>Hyperelastic Material>Plasticity

Membrane>Linear Elastic Material>Plasticity

Membrane>Nonlinear Elastic Material>Plasticity

Truss>Linear Elastic Material>Plasticity

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperplastic Material** node selected in the model tree:

Attributes>Plasticity

Creep

Use the **Creep** subnode to define the creep properties of the material model. This material model is available in the Solid Mechanics, Membrane, and Layered Shell interfaces, and can be used together with [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 <http://www.comsol.com/products/specifications/>.



See also [Creep and Viscoplasticity](#) in the *Structural Mechanics Theory* chapter.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Creep](#) node.

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 σ_{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 σ_{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_{shift} and the **Reference time** t_{ref} .
 - For **Strain hardening**, enter the **hardening exponent** m , the **Effective creep strain shift** $\varepsilon_{\text{shift}}$, and the **Reference time** t_{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** σ_{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** σ_{ref} . The default is 1 MPa.

Potential

For **Potential** enter the following settings:

- **Rate multiplier** η .
- **Creep potential** Q_{cr} .

Deviatoric

For **Deviatoric** enter the **Effective creep strain rate** ϵ_{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** ϵ_{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
Layered Shell>Linear Elastic Material>Creep
Membrane>Linear Elastic Material>Creep
Membrane>Nonlinear Elastic Material>Creep

Ribbon

Physics tab with **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 and Membrane interfaces, and can be used together with [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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Viscoplasticity](#) node.

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** ξ .
- **Stress sensitivity** m .
- **Deformation resistance saturation coefficient** s_0 .
- **Deformation resistance initial value** s_{init} .
- **Hardening constant** h_0 .
- **Hardening sensitivity** α .
- **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** σ_{ref} . The default is 1 MPa.
- **Stress exponent** n .

Yield Function F

The **Yield function F** defines the limit of the elastic regime $F(\sigma, \sigma_{\text{ys}}) \leq 0$.

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 non associated 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** $\sigma_{\text{ys}0ij}$ 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** (the default), **von Mises**, or **User defined** (non associated). Enter a **User defined** value in the Q field as needed.

Initial Yield Stress

The default **Initial yield stress** $\sigma_{\text{ys}0}$ 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** $E_{T\text{iso}}$ uses values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified as hardening occurs, and it is related to the effective viscoplastic strain ϵ_{vpe} as

$$\sigma_{ys} = \sigma_{ys0} + E_{\text{iso}}\epsilon_{vpe} \quad \text{with} \quad \frac{1}{E_{\text{iso}}} = \frac{1}{E_{T\text{iso}}} - \frac{1}{E}$$

For the linear isotropic hardening model, the yield stress increases proportionally to the effective viscoplastic strain ϵ_{vpe} . The Young's modulus E is taken from the elastic material properties.

- Select **Ludwik** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the power-law

$$\sigma_{ys} = \sigma_{ys0} + k\epsilon_{vpe}^n$$

the **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.

- For **Swift** nonlinear isotropic hardening, the **Reference strain** ϵ_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified by the power-law

$$\sigma_{ys} = \sigma_{ys0} \left(1 + \frac{\epsilon_{vpe}}{\epsilon_0} \right)^n$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

$$\sigma_{ys} = \sigma_{ys0} + \sigma_{\text{sat}} \left(1 - e^{-\beta\epsilon_{vpe}} \right)$$

the **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.

- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_{∞} , the **Saturation coefficient** m , and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{ys} = \sigma_{ys0} + (\sigma_{\infty} - \sigma_{ys0}) \left(1 - e^{-m \varepsilon_{vpe}^n}\right)$$

- For **User defined** isotropic hardening the **Hardening function** $\sigma_h(\varepsilon_{vpe})$ uses values **From material**. The yield level σ_{ys} is modified as

$$\sigma_{ys} = \sigma_{ys0} + \sigma_h(\varepsilon_{vpe})$$

This definition implies that the hardening function $\sigma_h(\varepsilon_{vpe})$ must be zero at zero viscoplastic strain. In other words, $\sigma_{ys} = \sigma_{ys0}$ when $\varepsilon_{vpe} = 0$. With this option it is possible to fit nonlinear hardening curves. The hardening function can depend on more variables than the effective viscoplastic strain, for example the temperature.

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 σ_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 σ_b is collinear to the viscoplastic strain tensor ε_{vp} .

- If **Armstrong-Frederick** is selected from the list, the default **Kinematic hardening modulus** C_k and **Kinematic hardening parameter** γ_k use values **From material**. These parameters are used to calculate the back stress σ_b from the rate equation

$$\dot{\sigma}_b = \frac{2}{3} C_k \dot{\varepsilon}_{vp} - \gamma_k \dot{\varepsilon}_{vpe} \sigma_b$$

This is 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:

$$\dot{\sigma}_{b,i} = \frac{2}{3}C_i\dot{\epsilon}_{vp} - \gamma_i\dot{\epsilon}_{vpe}\sigma_{b,i}$$

For each **Branch** row, enter C_i (the hardening modulus of the branch i) in the **Hardening modulus (Pa)** column and γ_i (the hardening parameter of the branch i) in the **Hardening parameter (1)** column.

Use the **Add** button () and the **Delete** button () to add or delete a row in the table. Use the **Load from file** button () and the **Save to file** button () to load and store data for the branches in a text file with three space-separated columns (from left to right): the branch number, the hardening modulus for that branch, and the hardening parameter for that branch.

The total back stress σ_b is then computed from the sum

$$\sigma_b = \frac{2}{3}C_0\epsilon_{vp} + \sum_{i=1}^{LV} \sigma_{b,i}$$

Perzyna

For **Perzyna** enter the following settings:

- **Viscoplastic rate coefficient** A .
- **Reference creep stress** σ_{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

Layered Shell>Linear Elastic Material>Viscoplasticity

Membrane>Linear Elastic Material>Viscoplasticity

Membrane>Nonlinear Elastic Material>Viscoplasticity

Ribbon

Physics tab with **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.

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**, or **FKM-GTN**.

Shima-Oyane

For **Shima-Oyane** enter the following data:

- **Initial yield stress** σ_{ys0} .
- **Shima-Oyane parameter** α .
- **Shima-Oyane parameter** γ .
- **Shima-Oyane parameter** m .
- **Initial void volume fraction** f_0 .

Gurson

For **Gurson** enter the following data:

- **Initial yield stress** σ_{ys0} .
- **Initial void volume fraction** f_0 .

Gurson-Tvergaard-Needleman

For **Gurson-Tvergaard-Needleman** enter the following data:

- **Initial yield stress** σ_{ys0} .
- **Tvergaard correction coefficient** q_1 .
- **Tvergaard correction coefficient** q_2 .
- **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** σ_{ys0} .
- **Initial void volume fraction** f_0 .
- **Maximum void volume fraction** f_{max} .

FKM-GTN

For **FKM-GTN** enter the following data:

- **Initial yield stress** σ_{ys0} .
- **Tvergaard correction coefficient** q_1 .
- **Tvergaard correction coefficient** q_2 .
- **Initial void volume fraction** f_0 .
- **Maximum void volume fraction** f_{max} .
- **Void volume fraction cut-off for GTN model** f_{GTN} .
- **Void volume fraction cut-off for FKM model** f_{FKM} .

Each of the material properties can either be defined **From material**, or as **User defined**. In the latter case, enter a value or an expression.



See also [Porous Plasticity](#) 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:

- [Drucker-Prager Criterion](#)
- [Mohr-Coulomb Criterion](#)
- [Matsuoka-Nakai Criterion](#)
- [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 α and k coefficients. Often material data is expressed in the parameters c and ϕ used in the Mohr-Coulomb model. You can choose to use these parameters instead. If so, select the **Match to Mohr-Coulomb criterion** check box (see [Mohr-Coulomb Criterion](#)). If this check box is selected, the default values for **Cohesion** c and the **Angle of internal friction** ϕ 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** ψ . 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** $\varepsilon_{pvol,max}$, 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** ϕ 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** ψ . 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** $\varepsilon_{pvol,max}$, 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** ϕ is taken **From material**.

If the **Match to Mohr-Coulomb criterion** check box is *not* selected, then the default **Matsuoka-Nakai mu coefficient** μ 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** ϕ . 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 CUT-OFF

If required, select the **Include tension cut-off** check box (see [Tension Cut-Off](#)). Enter a value or expression for the **Max tensile stress** σ_t . Use this to constrain the soil plasticity model with an extra yield surface, which limits the maximum principal stress.



- *Deep Excavation*: Application Library path **Geomechanics_Module/Soil/deep_excavation**
- *Flexible and Smooth Strip Footing on 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:

- [Bresler-Pister Criterion](#)
- [Willam-Warnke Criterion](#)
- [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** σ_t , **Uniaxial compressive strength** σ_c , and **Biaxial compressive strength** σ_b are taken **From material**.

Willam-Warnke

The defaults for the **Uniaxial tensile strength** σ_t , **Uniaxial compressive strength** σ_c , and **Biaxial compressive strength** σ_b are taken **From material**.

Ottosen

The defaults for the **Uniaxial tensile strength** σ_c , Ottosen's parameters a and b , **Size factor** k_1 , and **Shape factor** k_2 are taken **From material**.

TENSION CUT-OFF

If required, select the **Include tension cut-off** check box (see [Tension Cut-Off](#)). Enter a value or expression for the **Max tensile stress** σ_t . Use this to constraint the concrete model with an extra yield surface, which limits the maximum principal 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 criterion are described in the theory section:

- [Original Hoek-Brown Criterion](#)
- [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** σ_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** σ_c , **Geological strength index** GSI , **Disturbance factor** D , and **Intact rock parameter** m_i are taken **From material**.

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 <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

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_{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.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.

From the **Temperature** T list, select an existing temperature variable from a heat transfer interface (for example, **Temperature (ht)**), if any temperature variables exist. Select **User defined** to enter a value or expression for the temperature.



See also [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

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

where α is the secant coefficient of thermal expansion. α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

	<p>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.</p>
	<ul style="list-style-type: none"> • Thermoelastic Damping • Entropy and Thermoelasticity
	<p><i>Thermal Stresses in a Layered Plate</i>: Application Library path Structural_Mechanics_Module/Thermal-Structure_Interaction/layered_plate MEMS_Module/Actuators/layered_plate</p>

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Linear Elastic Material>Thermal Expansion
Solid Mechanics>Nonlinear Elastic Material>Thermal Expansion
Solid Mechanics>Hyperelastic Material>Thermal Expansion
Solid Mechanics>Piezoelectric Material>Thermal Expansion
Solid Mechanics>Cam-Clay Material>Thermal Expansion
Layered Shell>Linear Elastic Material>Thermal Expansion
Membrane>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**, **Piezoelectric Material**, or **Cam-Clay 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

$$\varepsilon_{\text{hs}} = \beta_{\text{h}} M_{\text{m}} (c_{\text{mo}} - c_{\text{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, M_{m} is the molar mass, c_{mo} is the moisture concentration, and $c_{\text{mo,ref}}$ is the strain-free reference concentration.

The **Hygroscopic Swelling** subnode is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Hygroscopic Swelling](#) node.

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 SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref} . This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as the **Concentration type**, enter also the **Molar mass** of the fluid, M_{m} . The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined**, select **Isotropic**, **Diagonal**, or **Symmetric** from the list to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h} . The default value for the **User defined** case is $1.5\text{e-}4 \text{ m}^3/\text{kg}$. 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

Membrane>Linear Elastic Material>Hygroscopic Swelling

Membrane>Nonlinear Elastic Material>Hygroscopic Swelling

Membrane>Hyperelastic Material>Hygroscopic Swelling

Ribbon

Physics tab with **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 <http://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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Initial Stress and Strain](#) 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. The given initial stresses and strains are interpreted in this system.

INITIAL STRESS AND STRAIN

Enter values or expressions for the **Initial stress** \mathbf{S}_0 and **Initial strain** ϵ_0 . For both, enter the diagonal and off-diagonal components (based on space dimension):

- For a 3D **Initial stress** model, diagonal components \mathbf{S}_{0x} , \mathbf{S}_{0y} , and \mathbf{S}_{0z} and off-diagonal components \mathbf{S}_{0xy} , \mathbf{S}_{0yz} , and \mathbf{S}_{0xz} , for example.
- For a 3D **Initial strain** model, diagonal components ϵ_{0x} , ϵ_{0y} , and ϵ_{0z} and off-diagonal components ϵ_{0xy} , ϵ_{0yz} , and ϵ_{0xz} , for example.

In a geometrically nonlinear analysis, the stresses should be interpreted as Second Piola-Kirchhoff stresses, and the strains should be interpreted as Green-Lagrange strains.



- For details about initial stresses and strains, see [Inelastic Strain Contributions](#) and [Initial 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

Solid Mechanics>Linear Elastic Material>Initial Stress and Strain

Solid Mechanics>Nonlinear Elastic Material>Initial Stress and Strain

Solid Mechanics>Piezoelectric Material>Initial Stress and Strain

Solid Mechanics>Cam-Clay Material>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**, **Piezoelectric Material**, or **Cam-Clay Material** 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 <http://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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the **External Stress** node.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor (Material)**, **Stress tensor (Spatial)** or **Pore pressure**. The option **Pore pressure** is not available in the Layered Shell interface.

- 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** drop-down list will contain all stress tensors announced by any physics interface, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** S_{ext} as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type** — **Add to stress tensor** 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** σ_{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_{ref} , which is the pressure level at which the pore pressure does not give any stress contribution. Enter the **Biot-Willis coefficient** a_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.



- 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.



- For theory, see [External Stress](#).
- For details about the different stress measures, see [Defining Stress](#).



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>Hyperelastic Material>External Stress
Layered Shell>Linear Elastic Material>External Stress
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

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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [External Strain](#) node.

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**.

Strain tensor

For **Strain tensor**, enter an inelastic strain contribution ϵ_{ext} . From the **Strain tensor** list, you can choose **User defined**, or any strain tensor 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_{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_{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 <http://www.comsol.com/products/specifications/>).

- Selecting a tensor announced by the same physics interface as where the **External Strain** node is added, may result in an error ('Circular variable dependency detected'). This operation is usually meaningless.
- You can use a tensor from the same physics interface, but from a previous solution step. Select the **User defined** input type and enter expressions where the `withsol` operator is used to point to the intended solution.
- If you select **Deformation gradient** form the **Strain input** list, and the **Additive strain decomposition** check-box is selected in the parent material node during a geometrically nonlinear study step, the Green-Lagrange strain tensor is computed from

$$\varepsilon_{\text{ext}} = \frac{1}{2}(\mathbf{F}_{\text{ext}}^T \mathbf{F}_{\text{ext}} - \mathbf{I})$$

- If you select **Deformation gradient** form the **Strain input** list when engineering strains are expected, the engineering strain tensor is computed from

$$\varepsilon_{\text{ext}} = \frac{1}{2}(\mathbf{F}_{\text{ext}} + \mathbf{F}_{\text{ext}}^T) - \mathbf{I}$$

This is the case ff 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** form the **Strain input** list, and a the parent material node operates with multiplicative strain decomposition, the external strain is converted into a deformation gradient using the infinitesimal strain assumption

$$\mathbf{F}_{\text{ext}} = \mathbf{I} + \varepsilon_{\text{ext}}$$



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

Layered Shell>Linear Elastic Material>External Strain

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** 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 can be used in combination with **Linear Elastic Material**.

DAMAGE

Select the type of **Damage model** — **Scalar damage** or **Mazars damage for concrete**.

Scalar damage

Select the type of **Equivalent strain** — **Rankine**, **Smooth Rankine**, **Norm of elastic strain tensor**, or **User defined**.

The **Activate damage in compression** check box is available for the **Rankine**, **Smooth Rankine**, or **User defined** equivalent strain, 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**, or **User defined**.

- For **Linear strain softening** or **Exponential strain softening** enter the **Tensile strength** σ_{ts} , the default is to take the value **From material**. 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** ε_f accordingly.

Mazars damage for concrete

Select the type of **Equivalent strain** — **Mazars**, **Modified Mazars**, or **User defined**.

Enter the **Shear exponent** β , the default is set to 1.06.

Select the **Tensile damage evolution** — **Linear strain softening**, **Exponential strain softening**, **Mazars damage evolution function**, or **User defined**.

- For **Linear strain softening** or **Exponential strain softening** enter the **Tensile strength** σ_{ts} , the default is to take the value **From material**. Also select the type of **Tensile strain softening** and enter the **Fracture energy per area** G_{ft} , the **Fracture energy per volume** g_{ft} , or the **Strain softening parameter** ε_{ft} accordingly.
- For **Mazars damage evolution function**, enter the **Tensile strain threshold** ε_{0t} , and the **Tensile damage evolution parameters** A_t and B_t .
- For **User defined**, enter an expression for the **Tensile damage evolution** function $d_t(\kappa)$.

Select the **Compressive damage evolution** — **Mazars damage evolution function** or **User defined**.

- For **Mazars damage evolution function**, enter the **Compressive strain threshold** ε_{0c} , and the **Compressive damage evolution parameters** A_c and B_c .
- For **User defined**, enter an expression for the **Compressive damage evolution** function $d_c(\kappa)$.

Regularization

Select the type of **Regularization method** — **None**, **Crack band**, or **Implicit gradient**.

- 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} .

ADVANCED

Enter the **Maximum damage**. The default value is 0.99999.

The **Damage** subnode is only available with some COMSOL products (see <http://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

Ribbon

Physics tab with **Linear Elastic Material**, node selected in the model tree:

Attributes>Damage

Activation

Use the **Activation** subnode to activate or de-activate 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 removed.

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 <http://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

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.

The **Safety** node is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>)

LAYER SELECTION

This section is only present when working with layered shells.



For a description of this section, see

- [Safety](#) in the documentation for the Layered Shell interface.
- [Safety](#) in the documentation for the Shell interface.

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	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

TABLE 4-3: AVAILABLE FAILURE CRITERIA BY PHYSICS INTERFACE

CRITERION	Solid Mechanics	SHELL, PLATE	LAYERED SHELL	MEMBRANE	BEAM	TRUSS
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	-	-
Tsai-Wu Orthotropic	X	X	X	X	-	-
Tsai-Wu Anisotropic	X	X	X	X	-	-
User defined	X	X	X	X	X	X

- When **Failure Criterion** is **von Mises Isotropic**, enter **Tensile strength** σ_{ts} .
- When **Failure Criterion** is **Tresca Isotropic**, enter **Tensile strength** σ_{ts} .
- When **Failure Criterion** is **Rankine Isotropic**, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs} .
- When **Failure Criterion** is **St. Venant Isotropic**, enter **Ultimate tensile strain** ϵ_{ts} and **Ultimate compressive strain** ϵ_{cs} .
- When **Failure Criterion** is **Mohr-Coulomb Isotropic**, select **Material parameters** — **Cohesion and angle of friction** or **Tensile and compressive strengths** to determine the type of input data.
 When **Cohesion and angle of friction** is used, enter **Cohesion** c and **Angle of internal friction** ϕ .
 When **Tensile and compressive strengths** is used, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs} .

In either case, you can select **Include elliptic cap** to limit the allowed compressive stress. When selected, enter the **Elliptic cap parameters** p_a and p_b .

- When **Failure Criterion** is **Drucker-Prager Isotropic**, select **Material parameters — Drucker-Prager parameters, Tensile and compressive strengths**, or **Mohr-Coulomb parameters** to determine the type of input data.
When **Drucker-Prager parameters** is used, enter **Drucker-Prager alpha coefficient** α and **Drucker-Prager k coefficient** k .
When **Tensile and compressive strengths** is used, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs} .
When **Mohr-Coulomb parameters** is used, enter **Cohesion** c and **Angle of internal friction** ϕ .
In either case, you can select **Include elliptic cap** to limit the allowed compressive stress. When selected, enter the **Elliptic cap parameters** p_a and p_b .
- When **Failure Criterion** is **Bresler-Pister Isotropic**, enter **Tensile strength** σ_{ts} , **Compressive strength** σ_{cs} , and **Biaxial compressive strength** σ_{bc} .
- When **Failure Criterion** is **Willam-Warnke Isotropic**, enter **Tensile strength** σ_{ts} , **Compressive strength** σ_{cs} , and **Biaxial compressive strength** σ_{bc} .
- When **Failure Criterion** is **Ottosen Isotropic**, enter the **Compressive strength** σ_{cs} , **Ottosen parameters** a and b , the **Size factor** k_1 , and the **Shape factor** k_2 .
- When **Failure Criterion** is **Jenkins Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Waddoups Orthotropic**, enter **Ultimate tensile strains** ϵ_{ts} , **Ultimate compressive strains** ϵ_{cs} , and **Ultimate shear strains** γ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Azzi-Tsai-Hill Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Norris Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Modified Tsai-Hill Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.

- When **Failure Criterion** is **Tsai-Hill Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Hoffman Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- When **Failure Criterion** is **Tsai-Wu Orthotropic**, enter **Tensile strengths** σ_{ts} , **Compressive strengths** σ_{cs} , and **Shear strengths** σ_{ss} . All entries have three components, related to the principal axes of orthotropy.
- 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** $s_f(S)$ respectively. 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>Safety
Solid Mechanics>Nonlinear Elastic Material>Safety
Shell>Linear Elastic Material>Safety
Layered Shell>Linear Elastic Material>Safety
Plate>Linear Elastic Material>Safety
Membrane>Linear Elastic Material>Safety
Membrane>Nonlinear Elastic Material>Safety
Beam>Linear Elastic Material>Safety
Truss>Linear Elastic Material>Safety

Ribbon

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic Material** node selected in the model builder tree:

Attributes>External Strain

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 Damping](#)
- [Anisotropic Loss Factor Damping](#)
- [Orthotropic Loss Factor Damping](#)
- [Viscous Damping](#)

The available damping models differs between various COMSOL products (see <http://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
Rayleigh damping	All	All	Time domain, frequency domain
Isotropic loss factor	All	All	Frequency domain
Anisotropic loss factor	All, except Beam and Truss	Linear Elastic	Frequency domain

TABLE 4-4: AVAILABLE DAMPING TYPES

DAMPING TYPE	PHYSICS INTERFACES	MATERIAL MODELS	STUDY TYPES
Orthotropic loss factor	All, except Beam and Truss	Orthotropic Linear Elastic	Frequency domain
Viscous damping	Solid Mechanics, Membrane, Multibody Dynamics	All	Time domain, frequency domain

LAYER SELECTION

This section is only present when working with layered shells.



For a description of this section, see

- [Damping](#) in the documentation for the Layered Shell interface
- [Damping](#) in the documentation for the Shell interface

DAMPING SETTINGS

Select a **Damping type**, and enter settings depending on the type.

Rayleigh Damping

In this damping model, the damping parameter ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{dM}m + \beta_{dK}k$$

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* α_{dM} and the *stiffness damping parameter* β_{dM} .

Select **Input parameters** — **Alpha and beta** — to enter the damping parameters explicitly, or **Damping ratios** to derive the damping parameters from the relative damping at two frequencies.

When **Alpha and beta** is selected, the **Mass damping parameter** α_{dM} and the **Stiffness damping parameter** β_{dK} .

When **Damping ratios** is selected, enter two pairs of frequencies, f_1 and f_2 , and the corresponding damping ratios ζ_1 and ζ_2 at these frequencies. The Rayleigh damping parameters are computed as

$$\alpha_{\text{dM}} = 4\pi f_1 f_2 \frac{\zeta_1 f_2 - \zeta_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{\text{dK}} = \frac{\zeta_2 f_2 - \zeta_1 f_1}{\pi(f_2^2 - f_1^2)}$$

Isotropic Loss Factor Damping

The isotropic loss factor damping is described by the single isotropic loss factor η_s , which acts on all entries in the elastic constitutive matrix. It can be used for isotropic, orthotropic, and anisotropic materials.

When **Isotropic loss factor** is selected, use the **Isotropic structural loss factor** list to select the way to enter η_s . The default is to take the value **From material**. For **User defined**, enter another value or expression.

Anisotropic Loss Factor Damping

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 η_s or by a symmetric anisotropic 6-by-6 loss factor matrix η_D or η_{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 η_D or η_{DV_0} . The default is to take the values **From material**. For **User defined** enter the components of η_D or η_{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 η_D using the standard notation.

Orthotropic Loss Factor Damping

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 Damping](#).
- Anisotropic, as described under [Anisotropic Loss Factor Damping](#).
- 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 η_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 η_G or η_{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** η_b and **Shear viscosity** η_v to model damping caused by volume change and deformation, respectively.



- 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
Layered Shell>Linear Elastic Material>Damping
Membrane>Linear Elastic Material>Damping
Membrane>Nonlinear Elastic Material>Damping
Membrane>Hyperelastic Material>Damping
Truss>Linear Elastic Material>Damping
Multibody Dynamics>Linear Elastic Material>Damping

Ribbon

Physics tab with **Linear Elastic Material**, **Hyperelastic Material**, or **Nonlinear Elastic 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 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** (the default), **Isotropic loss factor**, or **Rayleigh damping**.

Loss Factor for cE (Piezoelectric Material)

By default the **Loss factor for elasticity matrix cE** η_{cE} uses values **From material**. For **User defined** enter other values or expressions in the matrix.

Loss Factor for sE (Piezoelectric Material)

This choice is only available if the **Constitutive relation** in the parent node is **Strain-charge form**. By default the **Loss factor for compliance matrix sE** η_{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** η_{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** η_s uses values **From material**. For **User defined** enter another value or expression.

Rayleigh Damping

Enter the **Mass damping parameter** α_{dM} and the **Stiffness damping parameter** β_{dK} .

In this damping model, the damping parameter ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{dM}m + \beta_{dK}k$$

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* α_{dM} and the stiffness damping parameter β_{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

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** (the default) or **Rayleigh damping**.

- For **Loss factor for e** select a **Loss factor for coupling matrix e** from the list. Select **User defined** to enter values or expressions for η_e in a 3-by-6 matrix.
- For **Rayleigh damping** enter a **Stiffness damping parameter** β_{dC} .



See also [Piezoelectric Losses](#).

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>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 ϵS** (the default) or **Dispersion**.

For **Loss factor for ϵS** select a **Loss factor for electrical permittivity ϵS** . Select **From material** (the default) 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 η_{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** τ_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

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 a Eigenfrequency or Frequency response study.

CONDUCTION CURRENT

By default, the **Electrical conductivity** σ_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 **Anisotropic** depending on the characteristics of the electrical conductivity, and then enter values or expressions for the **Electrical conductivity** σ_e in the field or matrix.
- For **Linearized resistivity** the default **Reference temperature** T_0 , and **Resistivity temperature coefficient** α_r , and **Reference resistivity** ρ_0 are taken **From material**, which means that the values are taken from the domain (or boundary) material. T is the current temperature, which can be a value that is specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field appears in the **Model Inputs** section.



See also [Piezoelectric Losses](#).

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Piezoelectric Material>Conduction Loss (Time-Harmonic)
Electrostatics>Charge Conservation>Conduction Loss (Time-Harmonic)

Ribbon

Physics tab with **Piezoelectric Material** or **Charge Conservation** node selected in the model tree:

Attributes>Conduction Loss (Time-Harmonic)

External Stress-Strain Relation

The **External Stress-Strain Relation** is a special type of material model where the computation of 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 <http://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 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*
-

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Material Models>External Stress-Strain Relation

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>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 <http://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.

LAYER SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the [Rigid Domain](#) node.

DENSITY

The default **Density** ρ is taken **From material**. In this case the material assignment for the domain supplies the mass density. For **User defined** enter another value or expression.

If any material in the model has a temperature dependent mass density, and **From material** is selected, the **Volume reference temperature** list will appear in the **Model Input** section. As a default, the value of T_{ref} is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the reference temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature; 293.15 K.

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** in the current component, having the same selection as in the current node.



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](#).
- [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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
---	------------------------------------

	<i>Modeling Rigid Bodies</i> : Application Library path Structural_Mechanics_Module/Connectors_and_Mechanisms/rigid_domain
---	--

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** Ω
- **Angle of rotation** ϕ .
- **Angular velocity** ω (3D components) and $\partial\phi/\partial t$ (2D components)

LOCATION IN USER INTERFACE

This node is automatically added when the Rigid 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](#).

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.

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

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.



The **Fixed Constraint** and **Prescribed Displacement/Rotation** subnodes cannot be combined for a Rigid Domain.

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** (the default), **Constrained rotation**, or **Prescribed rotation**.



For 2D components, and if **Prescribed rotation** is selected, enter a value or expression for the **Angle of rotation** ϕ_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** Ω . Then enter a value or expression for the **Angle of rotation** ϕ_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.
- 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

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.

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

Enter values or expressions for the components of the **Applied force \mathbf{F}** .



- 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

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.

APPLIED MOMENT

Enter values or expressions for the components of the **Applied moment**.

- For 3D components, this is the **x**, **y**, and **z** components for **M**.
- For 2D components, this is for 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

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** (the default), **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

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 **Anisotropic**. 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 **Anisotropic**. 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 **Anisotropic**, then enter values or expressions in the table for the **Loss factor for spring** $\eta_{u,k}$ or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

All settings in the **Rotational Loss Factor Damping** section are analogous to the corresponding settings in the **Loss Factor** 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 **Anisotropic**, then enter values or expressions for the damping constants d_{ij} 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 the **Viscous Rotational Damping** 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

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 interfaces 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 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

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>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](#).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Prescribed Displacement](#) and [Prescribed Displacement, Interface](#) nodes.

PRESCRIBED DISPLACEMENT

Define the prescribed displacements using a **Standard notation** (the default) 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 **Anisotropic** 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** button () and select **Advanced Physics Options**.



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** drop-down 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 <http://www.comsol.com/products/specifications/>).



For details about prescribed velocities and accelerations, see [Prescribed Displacements, Velocities, and Accelerations](#).

LAYER SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the [Prescribed Velocity](#) and [Prescribed Velocity, Interface](#) nodes.

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 a 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 <http://www.comsol.com/products/specifications/>).



For details about prescribed velocities and accelerations, see [Prescribed Displacements, Velocities, and Accelerations](#).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Prescribed Acceleration](#) and [Prescribed Acceleration, Interface](#) nodes.

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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Fixed Constraint](#) and [Fixed Constraint, Interface](#) nodes.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.



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.



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

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

The **Thermal Expansion** subnode is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

This section is only present in the Layered Shell interface, where it is described in the documentation for the [Thermal Expansion \(for Constraints\)](#) node.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from domain** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation. For this reason, it is only possible to inherit from a **Thermal Expansion** node under a material model, not from the **Thermal Expansion** multiphysics coupling.
- The virtual surrounding material has the same thermal expansion as the domain itself.

The **Inherit from domain** 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** α . 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 α . 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_{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** button () and select **Advanced Physics Options**.



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 <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

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 drop-down 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** button () and select **Advanced Physics Options**.



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

Ribbon

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 <http://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 drop-down 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** button () and select **Advanced Physics Options**.



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

Ribbon

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 <http://www.comsol.com/products/specifications/>).



In a geometrically nonlinear analysis, large rotations must not occur at the antisymmetry plane because this causes artificial straining.

LAYER SELECTION

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** button () and select **Advanced Physics Options**.



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>Antisymmetry

Layered Shell>Antisymmetry

Membrane>More Constraints>Antisymmetry

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries>More Constraints>Antisymmetry

Physics tab with **Layered Shell** selected:

Edges>More Constraints>Antisymmetry

Physics tab with **Membrane** 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.

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.

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Rigid Motion Suppression](#) node.

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none">• Constraint Reaction Terms• Weak Constraints• Constraint Settings
	<ul style="list-style-type: none">• Rigid Body Motion in the Structural Mechanics Modeling chapter.• Rigid Motion Suppression Boundary Condition in the Structural Mechanics Theory chapter.
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.</p>

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.

LAYER SELECTION

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** (the default), **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.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit volume	F_V	N/m^3	domains	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit area	F_A	N/m ²	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 <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

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 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 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 **Rotating Frame** node is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

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**, or **Revolutions per time**. Then enter a value as needed for **Angular velocity magnitude** Ω , or **Revolutions per time**.

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.

LAYER SELECTION

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** (the default), **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 body load as force per unit volume is then the value of F divided by the thickness.

- For **Total force**, COMSOL Multiphysics then divides the total force by the area of the surfaces where the load is active.
- 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.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit area	F_A	N/m ²	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.

LAYER SELECTION

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** (the default) 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}_{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.

LAYER SELECTION

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** (the default) 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 <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Spring Foundation](#) or [Spring Foundation, Interface](#) nodes.

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 4-5](#) 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 **Anisotropic**. 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 4-6](#). 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 **Layer thickness**, d_s , to specify the physical thickness of the elastic layer.

TABLE 4-5: 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

TABLE 4-5: SPRING TYPES FOR THE SPRING FOUNDATION FEATURE

SPRING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION
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 4-6: 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	$\nu =$	ν	$\frac{E}{2G} - 1$	$\frac{1}{2} \left(1 - \frac{3G}{3K + G} \right)$
Bulk modulus	$K =$	$\frac{E}{3(1 - 2\nu)}$	$\frac{EG}{3(3G - E)}$	K
Shear modulus	$G =$	$\frac{E}{2(1 + \nu)}$	G	G

ROTATIONAL SPRING

This section is available for the Beam interface. All settings in the **Rotational Spring** section 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** η_s which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic**, then enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

This section is available for the Beam interface. All settings in the **Rotational Loss Factor Damping** section are analogous to the corresponding settings in the **Loss Factor** section.

VISCOUS DAMPING

Select the **Damping type** using [Table 4-7](#) 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 **Anisotropic**. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 4-7: 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 interface. All settings in the **Rotational Viscous Damping** 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

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** 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 <http://www.comsol.com/products/specifications/>).

LAYER 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** 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 4-8](#) 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 **Anisotropic**. 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 4-6](#). 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 **Layer thickness**, d_s , to specify the physical thickness of the elastic layer.

TABLE 4-8: SPRING TYPES FOR THE THIN ELASTIC LAYER FEATURE

SPRING TYPE	VARIABLE	SI UNITS	SPACE DIMENSION
Total spring constant	k_{tot}	N/m	3D, 2D, and 2D axisymmetric
Spring constant per unit area	k_A	$N/(m \cdot m)^2$	3D, 2D, and 2D axisymmetric
Spring constant per unit length	k_L	$N/(m \cdot m)$	2D
Total force as function of extension	F_{tot}	N	3D, 2D, and 2D axisymmetric
Force per area as function of extension	F_A	N/m^2	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

LOSS FACTOR DAMPING

From the **Loss factor type** list, select **Scalar (Same for all components)** or **Individual components**.

- For **Scalar (Same for all components)** enter a single **Loss factor for spring** η_s which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS DAMPING

Select the **Damping type** using [Table 4-9](#) 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 Anisotropic. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 4-9: 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 ²)	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.

LAYER SELECTION

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

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 <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Added Mass](#) and [Added Mass, Interface](#) nodes.

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 4-10](#) 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 4-10: AVAILABLE MASS TYPES BASED ON GEOMETRIC ENTITY

MASS TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Mass per unit volume	ρ_V	kg/m^3	domains, boundaries
Mass per unit area	ρ_A	kg/m^2	domains, boundaries, edges
Mass per unit length	ρ_L	kg/m	edges, points
Total mass	m	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 <http://www.comsol.com/products/specifications/>).

The two sets of boundaries between which there is a periodicity condition are called the *source* and *destination*, respectively. 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 based 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** chose how to define the sector angle that the geometry represents using **Sector angle**. If **Automatic** is selected, the program attempts to find out how many full repetitions of the geometry there will be on a full revolution. If **User defined** is selected, enter a value for the sector angle θ_S . In both cases, also enter an **Azimuthal mode number** for the mode to be studied. It can vary from 0 to $N/2$, where N is the total number of sectors on a full revolution.
- For **User defined** select the 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 $u_i(\mathbf{x}_d) = -u_i(\mathbf{x}_s)$, respectively. 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** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)

ORIENTATION OF SOURCE

For information about the **Orientation of Source** 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 representing a large repetitive structure. Periodic boundary conditions will be used on the outer boundaries of this unit cell.

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**.

The **Average strain** periodicity type allows to derive homogenized elastic properties of media with a periodic structure, 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 ϵ_{avg} . In a geometrically nonlinear analysis, the average strains are interpreted as Green-Lagrange strains.

Select **Calculate average properties** — **None**, **Elasticity matrix**, **Standard**, or **Elasticity matrix, Voigt**.

The **Average stress** periodicity type allows to derive the homogenized compliance matrix of media with a periodic structure. Enter values or expressions for the components of the **Average stress** tensor σ_{avg} . In a geometrically nonlinear analysis, the stresses are interpreted as Second Piola-Kirchhoff stresses.

Select **Calculate average properties** — **None**, **Compliance matrix**, **Standard**, or **Compliance matrix, Voigt**.

With the **Mixed** periodicity type, enter either the average strain or average stress tensor components.

For all the options of **Periodicity type** enter the **Cell volume** — **Solid**, **Void volume fraction**, or **User defined**. The **Solid** option computes the volume of the RVE from the domains selected by the **Cell Periodicity** feature. The **Void volume fraction** option, which default is set to 0, scales the RVE volume when there are voids not selected as domains. The **User defined** option, which default is 1 m^3 , allows to enter the volume of the RVE directly.

The variable `item.vol` consider the whole volume of the domains selected under **Cell Periodicity** feature when computing average strains and stresses, adjust for porosity or scale the volume if needed.

LOAD GROUP, MATERIAL AND STUDY GENERATION

This section is available when **Periodicity Type** is **Average Strain** or **Average Stress**.

Click the **Create** button to add the features needed to perform a complete evaluation of the average material stiffness for a homogenized material. The following changes will be done 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 with 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.
- It is possible to compute the macroscopic elasticity matrix from the unit cell when the **Periodicity Type** is **Average Strain**. Select the **Generate material** check box to create a new global material, **Homogeneous Material 1**, containing the anisotropic data.



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*.



See also

- *Micromechanical Model of a Composite*: Application Library path **Structural_Mechanics_Module/Tutorials/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** button () and select **Advanced Physics Options**.



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 than 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 <http://www.comsol.com/products/specifications/>).

DAMPING

Select a **Damping type** — **P and S waves** (the default) or **User defined**. For **User defined** enter values or expressions for the **Mechanical impedance** \mathbf{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 <http://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](#).

Also see [Theory for the Thin-Film Flow Interfaces](#) as the **Thin-Film Damping** boundary condition is equivalent to an appropriately coupled Thin-Film Flow interface — so 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.

The **Rigid Connector** node is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>).

LAYER SELECTION

This section is only present in the in the Layered Shell interface, where it is described in the documentation for the [Rigid Connector](#) and [Rigid Connector, Interface](#) nodes.

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 can not 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 can not 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, since 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 point level in 3D at least three points, not located on a straight line, are needed.
- At point level in 2D at least two points are needed.
- At edge level in 3D, a single straight edge is not sufficient since it can act as a hinge.

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 non-linear 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 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 rigid connector 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**, 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_{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** ϕ_0 . For 3D components also enter an **Axis of rotation** Ω 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 are coupled to 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**.



- 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](#)



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](#)

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

LAYER SELECTION

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** α . 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 α . 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_{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 it is also a moment contribution. The directions of the force is fixed in space and does not follow the rotation of the rigid connector.

COORDINATE SYSTEM SELECTION

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

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 direction of the moment is fixed in space and does not follow the rotation of the rigid connector.

COORDINATE SYSTEM SELECTION

The **Coordinate system** list contains any applicable coordinate systems that the model contains. Select a **Coordinate system** in which the directions of the moment is given.

In 2D, the moment is a scalar input, and is not affected by the coordinate system selection.

APPLIED MOMENT

Enter values or expressions for the components of the **Applied moment \mathbf{M}**

- For 3D components, this is the **x**, **y**, and **z** components for **\mathbf{M}** . The direction coordinate names can vary depending on the selected coordinate system.
- For 2D components, this is for the applied moment in the **z** direction **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** (the default), **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 **Anisotropic**. 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 **Anisotropic**. 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 **Anisotropic**, then enter values or expressions in the table for the **Loss factor for spring** $\eta_{u,k}$ or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

ROTATIONAL LOSS FACTOR DAMPING

All settings in the **Rotational Loss Factor Damping** section are analogous to the corresponding settings in the **Loss Factor** 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 **Anisotropic**, 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 the **Viscous Rotational Damping** 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 the addition of the Multibody Dynamics 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 non-linear 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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

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** α . 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 α . 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_{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

Beam Connection



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 <http://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** (the default), **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** (the default), **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



The **Shell Connection** node is obsolete, and has been superseded by the [Solid-Shell 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 <http://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** (the default), **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** (the default) or **Parallel**. For **Parallel**, select a **Distance evaluation** — **Shell properties** (the default), **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:

Boundaries>Connections>Shell Connection

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.

If friction is to be included in the modeling, add a [Friction](#) subnode.

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 <http://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 interfaces with displacement degrees of freedom in your model, only the last physics interface in the model tree can contain contact physics features since that is the physics interface in which the displacements controls the spatial frame.



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.



A **Contact** node will override all previous nodes in the Model 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.

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 PRESSURE METHOD

Select the algorithm used for computing the contact in the normal direction, **Augmented Lagrangian** (default) or **Penalty**.

Enter a value for **Characteristic stiffness** E_{char} . The default is `solid.Eeq`, the equivalent Young's modulus as defined by most materials. The characteristic stiffness (stored in a variable named `solid.<contact_tag>.E_char`) is used in expressions for the default penalty factors for both the **Augmented Lagrangian** and **Penalty** methods. It 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 `solid.Eeqv` is not defined by the material. This is the case for some user-defined materials.

PENALTY FACTOR

If **Contact Pressure Method** is **Augmented Lagrangian**, 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.



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 result. You can consider the penalty factor as a spring giving a resisting force if the boundaries in the contact pair have an interference. A larger value gives faster, but less stable, convergence to the state where there is no interference. A too large value can create convergence difficulties, particularly if the interference is large

In the penalty method, the penalty factor is the actual stiffness of a spring inserted between the boundaries in the contact pair.

If you select **Preset**, you have the choices to select **Tuned for** as **Stability** (default) and **Speed**. If the contact boundaries move toward each other, so that large interferences can be expected in the initial iterations, then is **Stability** is the better choice. In many models, where the contact state does not change much, using **Speed** gives significant performance improvements.

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**. The default value is 1. Increasing this factor gives a higher penalty factor. From **Use relaxation**, select **Always** (default), **Never**, or **Conditional**. When using relaxation, the penalty factor is decreased during the first iterations in each parameter or time step.

If relaxation is used, enter the **Initial Relaxation Factor**. The default is 0.005. This factor multiplies the penalty factor in the first iteration. Enter **Number of iterations with relaxation**. The default value is 4. The penalty factor is gradually increased up to its full value, which is used in the 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, you could 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 interference 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. Edit or use the default **Contact pressure penalty factor** p_n . The default value is $\min(1e-3 * 5^{niterCMP}, 1) * \text{solid}.\langle \text{contact_tag} \rangle.E_char / \text{solid}.hmin_dst$. The default value causes the penalty factor to be increased during the iterations and takes material stiffness and element size at the contact surface into account. The variable `solid.<contact_tag>.E_char` is the characteristic stiffness as defined in the **Contact Pressure Method** section, and `hmin_dst` is the minimum element size on the destination.

When **Contact Pressure Method** is **Penalty**, enter the **Contact pressure penalty factor** p_n . The default value is `solid.<contact_tag>.E_char / solid.hmin_dst`. 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 .

TRIGGER CUTBACK

This section is only available when **Contact Pressure Method** is **Augmented Lagrangian**. Select the **Trigger cutback** check box to enable additional control over solver cutbacks in Time Dependent study or Stationary study with a Parametric continuation solver. 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 does 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](#)
-

OFFSET AND ADJUSTMENT

Enter a value or expression for **Contact surface 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 **Contact surface 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 touch each other in the initial state. The gap is adjusted to zero before any offset is added.



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 any initial distance between source and destination boundaries.



[Offset and Adjustment](#)

INITIAL VALUES

This section is only available when **Contact Pressure Method** is **Augmented Lagrangian**. Enter an initial value for the **Contact pressure** T_n .



If not all the contacting parts are fully constrained, it is important to supply a value here in order to avoid singular problems.



Speed up the convergence by supplying a guess of the correct order of magnitude of the contact pressure.

DISCRETIZATION

This section is only available when **Contact Pressure Method** is **Augmented Lagrangian**. Select a shape-function order for **Contact pressure** — **Linear** (the default), **Quadratic**, **Cubic**, **Quartic**, or (in 2D) **Quintic**. This setting should usually not be changed. Selecting anything else than **Linear** requires that the solver sequence is modified manually because the lumped solver is then no longer optimal for the contact pressure update.

ADVANCED

To display this section, click the **Show** button () and select **Advanced Physics Options**.

When analyzing a multiphysics contact problem, the state of being in contact or not will be passed from the contact analysis in the structural mechanics interface to other participating physics interfaces. For numerical reasons, the gap value will however not be exactly zero even when the boundaries are in contact. A certain small positive value of the gap will thus be considered as being in contact. The default **Multiphysics contact tolerance** is **Automatic**. If you want to explicitly specify the limit of the gap considered as being in contact, select **Manual**, and enter the **Contact tolerance** Δ_{contact} .

Select **Add contact status to solver log** to get printout about the changes in the solver log window. Doing this will add extra dependent variables used for tracking the contact state on the destination boundaries.

If you have more than one contact pair selected in the **Contact** node, and the augmented Lagrangian method is used, the check box **Group contact variables in solver per pair** will be shown. When selected, a new **Lumped Step** will be generated in the segregated solver for each contact pair. This will not affect the solution, but will give more granular output in the convergence plots.

Select **Store energy variables** to get access to variables containing various energy quantities specific to the contact condition.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Pairs>Contact

Multibody Dynamics>Pairs>Contact

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Pairs>Contact

Friction

Using the **Friction** subnode, you can add friction to a **Contact** node.

The selection of the **Friction** node is the same as that of its parent **Contact** node. If you add more than one **Friction** node under the same parent, the last one will override all the preceding nodes. If **Friction** and **Adhesion** are present under the same **Contact** node, the friction settings will be ignored once the adhesion criterion is fulfilled.

FRICITION

Select a **Friction model** — **Static Coulomb friction** or **Exponential dynamic Coulomb friction**. The latter can only be used in a time dependent analysis, since the value of the frictional coefficient depends on the slip velocity. Enter the following data:

- **Static frictional coefficient** μ_{stat} to give the static coefficient of friction.
- **Cohesion sliding resistance** T_{cohe} . Supply a traction which must be overcome before sliding can occur.
- **Maximum tangential traction** $T_{t,\text{max}}$. 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.
- When **Exponential dynamic Coulomb friction** is selected, also enter the **Dynamic frictional coefficient** μ_{dyn} and the **Friction decay coefficient** α_{def} .

TANGENTIAL FORCE METHOD

Select the algorithm used for computing the tangential forces and sliding criteria, **Augmented Lagrangian** or **Penalty**. The method can be selected independently of the corresponding **Contact Pressure Method** setting in the parent **Contact** node.

PENALTY FACTOR

If **Tangential Force Method** is **Augmented Lagrangian**, select the type of **Penalty factor control** — **From parent** (default), **Preset**, **Manual tuning**, or **User defined**. The settings give access to an increasing level of detailed control of the penalty factor for the sliding.

In the default case the settings are taken from the **Contact** node to which the current **Friction** node is a child. For a description of the settings, see the documentation of **Penalty Factor** under **Contact**. The only difference is that the default value of the **User defined** penalty factor is

$\min(1e-3*5^{\text{niterCMP},1}) * \text{solid}.\langle \text{contact_tag} \rangle .E_char/3/\text{solid}.\text{hmin_dst}.$

If **Tangential Force Method** is **Penalty**, select the type of **Penalty factor control** — **From parent** or **User defined**. The choice **From parent** is only available if **Contact Pressure Method** is set to **Penalty** in the parent **Contact** node. In that case, the settings are taken from the **Contact** node to which the current **Friction** node is a child. When **User defined** is selected, enter **Tangential pressure force factor** p_t . The default value is $\text{solid}.\langle \text{contact_tag} \rangle .E_char/3/\text{solid}.\text{hmin_dst}.$

INITIAL VALUES

If **Tangential Force Method** is **Augmented Lagrangian**, 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** (the default) or **In contact**.

For **In contact** enter values or expressions for the **Previous mapped source coordinates** $\mathbf{x}_{m, \text{old}}$. 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 components in the material frame.

DISCRETIZATION

This section can only be displayed if **Tangential Force Method** is **Augmented Lagrangian**. Select a shape-function order for **Friction force** — **Linear** (the default), **Quadratic**, **Cubic**, **Quartic**, or (in 2D) **Quintic**. This setting should usually not be changed. Selecting anything else than **Linear** requires that the solver sequence is modified manually

because the lumped solver is then no longer optimal for the update of the tangential forces.

ADVANCED

To display this section, click the **Show** button () and select **Advanced Physics Options**.

For numerical reasons, the gap value will not be exactly zero even when the boundaries are in contact. A certain small positive value of the gap must thus be used to determine that the boundaries touch each other, so that friction forces can be introduced. Select a **Friction detection** — **Automatic** or **Manual**. For **Manual** enter an absolute value for the **Friction detection tolerance** Δ_{contact} , which is the gap when the friction becomes active.

You can also request that the total accumulated slip distance is stored by selecting the **Store accumulated slip** check-box.



The contact tolerance is the gap distance between the surfaces at which friction starts to act. The default value is 10^{-6} times smaller than the overall size of the model.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Contact>Friction

Multibody Dynamics>Contact>Friction

Ribbon

Physics tab with **Contact** selected in the Model Builder tree:

Attributes>Friction

Adhesion

Using the **Adhesion** subnode, you can add adhesion and decohesion 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. If you add more than one **Adhesion** node under the same parent, the last one will override all

the preceding nodes. If **Adhesion** and **Friction** are present under the same **Contact** node, the friction settings will be ignored once the adhesion criterion is fulfilled.



Adhesion is only available in the context menu and ribbon if

- **Contact Pressure Method** is set to **Penalty** in the **Contact** settings.
- **Offset penalty function** is not selected in the **Contact** settings.

An existing **Adhesion** node will be disabled if any of these settings are changed later.

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_{n0} , at which adhesion is initiated.

When the criterion is **Gap**, enter the distance between the source and destination boundaries δ_0 , at which adhesion is initiated. A negative value indicates that there must be a certain overlap before adhesion is activated.

When the criterion is **User Defined**, enter a boolean expression such that when it evaluates to true, adhesion is initiated.

When the criterion is **Always active**, the source and destination boundaries are in 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.

Select how to specify the **Normal stiffness** of the adhesive layer. The default is to use **From contact penalty factor**, in which case the stiffness is the same as the one given as **Contact pressure penalty factor** in the settings of the parent **Contact** node.

To use another stiffness, select **User defined**, and enter the stiffness in the normal direction K_p explicitly. This value is used in tension only; in compression the penalty factor is always used.

Select **Shear stiffness defined using** to be either **Normal to shear ratio** or **Adhesive Poisson's ratio**.

For **Normal to shear ratio**, enter the ratio between the shear stiffness and the normal stiffness n_{τ} .

For **Adhesive Poisson's ratio**, enter Poisson's ratio for the adhesive layer, ν , explicitly.

DECOHESION

Select a **Traction separation law**. The default is **No separation**, in which case no decohesion will occur.

For **Linear separation**, **Polynomial separation**, or **Multilinear separation** enter:

- **Tensile strength** σ_{Ic} . This is the peak stress in pure tension.
- **Shear strength** τ_{Ic} . This is the peak stress in pure shear.
- **Tensile energy release rate** G_{Ic} . This is the energy released during the whole decohesion process, in pure tension.
- **Shear energy release rate** G_{IIc} . This is the energy released during the whole decohesion process, in pure shear.

For **Linear separation** or **Polynomial separation**, select the **Failure criterion** to be either **Power law** or **Benzeggagh-Kenane**. In either case, enter the **Mode mixity exponent** η .

For **Multilinear separation** enter the **Shape factor** λ .



See also

- [Adhesion and Decohesion](#), and [Adhesion](#) in the Structural Mechanics Modeling chapter.
- [Adhesion](#) and [Decohesion](#) in the Structural Mechanics Theory chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Contact>Adhesion

Multibody Dynamics>Contact>Adhesion

Ribbon

Physics tab with **Contact** selected in the Model Builder tree:

Attributes>Adhesion

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 <http://www.comsol.com/products/specifications/>).

BOLT PRETENSION

Select a **Pretension type** — **Pretension force** (the default) 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 σ_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 an 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 internal 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 <http://www.comsol.com/products/specifications/>).

BOLT GEOMETRY

Enter the **Half thread angle**, α . The default angle is 30[deg], which is used in for example the ISO and UTS standards. For 2D axial symmetry, this is the only input required to describe the bolt geometry. For 3D, you need also to provide the following settings:

Enter the **Lead**, l . This is the axial distance a single thread travels after one full revolution of the bolt.

Select a **Handedness** — **Right-handed** (the default) or **Left-Handed**. Most bolts are right-handed.

Select a **Bolt geometry** — **Automatic** (the default) 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 center line 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** (the default), **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, `solid.btc1.Eequ/solid.btc1.hmin_dst`, is the ratio between the material stiffness and the element size.

Enter a value for the **Frictional coefficient**, μ .



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

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response the physics interface splits the harmonic load into two parameters:

- The amplitude, F , which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

LAYER SELECTION

This section is only present in the Layered Shell interface. See also [The Layer Selection Section](#) and the documentation for the corresponding parent nodes.

PHASE

Enter the components of **Load phase** ϕ in radians (for a pressure the load phase ϕ is a scalar value). Add [deg] to a phase value to specify it using degrees.



Typically the load magnitude is a real scalar value. If the load specified in the parent feature contains a phase (using a complex-valued expression), the software adds the phase from the **Phase** node to the phase already included in the load.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Body Load>Phase

Solid Mechanics>Boundary Load>Phase

Solid Mechanics>Edge Load>Phase

Solid Mechanics>Point Load>Phase

Solid Mechanics>Ring Load>Phase

Solid Mechanics>Point Load (on Axis)>Phase

Solid Mechanics>Rigid Connector>Applied Force>Phase

Solid Mechanics>Rigid Connector>Applied Moment>Phase

Solid Mechanics>Rigid 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.

LAYER SELECTION

This section is only present in the Layered Shell interface. See also [The Layer Selection Section](#) 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 <http://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

Important Variables in the Solid Mechanics Interface

In each node in the physics interface, a number of variables are defined. In [Table 4-11](#), 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 the 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 4-11: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.disp	Total displacement	All materials ¹	
solid.vel	Velocity magnitude	All materials ¹	
solid.u_tX	Velocity, X component	All materials ¹	
solid.acc	Acceleration magnitude	All materials ¹	
solid.u_ttX	Acceleration, X component	All materials ¹	
solid.eXX	Strain tensor, XX component	All materials ¹	Total (Green-Lagrange) strain in global directions. See also Deformation Measures .
solid.eI11	Strain tensor, local coordinate system, II component	All materials ¹	Total strain in the local directions.
solid.eeI11	Elastic strain tensor, local coordinate system, II component	All materials ¹	

TABLE 4-11: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.eeldev11	Deviatoric elastic strain tensor, local coordinate system, 11 component	All materials ¹	
solid.eiel11	Inelastic strain tensor, local coordinate system, 11 component	All materials ¹	See also Inelastic Strain Contributions .
solid.FdxX	Deformation gradient, xX component	All materials ¹	See also Deformation Measures .
solid.Fdlx1	Deformation gradient, local, x1 component	All materials ¹	
solid.J	Volume ratio	All materials ¹	Total relative volume change.
solid.Cel11	Elastic Cauchy-Green tensor, local coordinate system, 11 component	All materials ¹	
solid.Jel	Elastic volume ratio	All materials ¹	Elastic relative volume change.
solid.Ldxy	Rate of strain tensor, xy component	All materials ¹	See also Strain Rate and Spin .
solid.Lwxy	Spin tensor, xy component	All materials ¹	See also Strain Rate and Spin .
solid.SXX	Second Piola-Kirchhoff stress, XX component	All materials ¹	See also Defining Stress .
solid.S111	Second Piola-Kirchhoff stress, local coordinate system, 11 component	All materials ¹	
solid.SdevX	Deviatoric second Piola-Kirchhoff stress, X component	All material ¹	See also Invariants of the Stress Tensor .
solid.PxX	First Piola-Kirchhoff stress, xX component	All materials ¹	See also Defining Stress .

TABLE 4-11: IMPORTANT VARIABLES IN THE SOLID MECHANICS INTERFACE

VARIABLE	DESCRIPTION	DEFINED IN	COMMENTS
solid.sxy	Stress tensor, xy component	All materials ¹	Cauchy stress, See also Defining Stress .
solid.sdevx	Deviatoric stress tensor, x component	All materials ¹	Deviatoric Cauchy stress, See also Invariants of the Stress Tensor .
solid.mises	von Mises stress	All materials ¹	See also Invariants of the Stress Tensor .
solid.tresca	Tresca stress	All materials ¹	See also Invariants of the Stress Tensor .
solid.I1s	First principal invariant of stress	All materials ¹	See also Invariants of the Stress Tensor .
solid.I2s	Second principal invariant of stress	All materials ¹	See also Invariants of the Stress Tensor .
solid.I3s	Third principal invariant of stress	All materials ¹	See also Invariants of the Stress Tensor .
solid.II2s	Second invariant of stress deviator	All materials ¹	See also Invariants of the Stress Tensor .
solid.II3s	Third invariant of stress deviator	All materials ¹	See also Invariants of the Stress Tensor .
solid.RFx	Reaction force, x component	All materials ¹	
solid.RMx	Reaction moment, y component	All materials ¹	
solid.Tax	Traction (force/area), x component	All materials ¹	

¹ - 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:

- [Theory for Shell and Plate Interfaces](#)
- [The Shell and Plate Interfaces](#)

Theory for Shell and Plate Interfaces

The theory for the Shell and Plate interfaces is discussed in this section:

- [About Shells and Plates](#)
- [Theory Background for the Shell and Plate Interfaces](#)
- [Theory for FSDT Laminated Shell](#)
- [References for the Shell Interface](#)

About Shells and Plates

A *shell* is a thin-walled structure in 3D or 2D axisymmetry where a simple form is assumed for the variation of the displacement through the thickness. Using this approximation, it is possible to develop a model for the deformation that is more similar to a 2D plane stress condition than to a full 3D state.



For a shell to give accurate results it is important that the structure can really be described as being thin-walled. When modeling using shells you should in general model the faces at the midplane of the real geometry. You can use the offset setting if the midsurface of the shell does not coincide with the boundary of your 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 either 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.

To describe a shell, you provide its thickness, a possible offset, and the elastic material properties.

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, it has an MITC formulation where MITC means *mixed interpolation of tensorial components*. A general description of this element family is given in [Ref. 1](#).

The dependent variables 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.

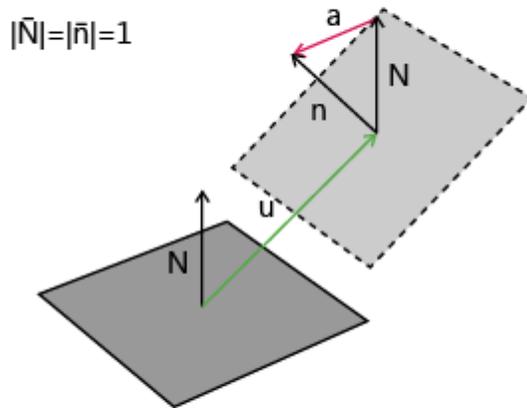


Figure 5-1: The degrees of freedom in the shell interface. \mathbf{N} is the normal vector in the original configuration and \mathbf{n} is the normal in the deformed state.

The degrees of freedom represent the displacements on the *reference surface*. The reference surface is the boundary where the shell element mesh is created. 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 ζ_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 R - Z plane.

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

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



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

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, ξ^i be element local (possibly nonorthogonal) coordinates with origin in the shell midsurface, and \mathbf{n} be the normal to the undeformed midsurface. The thickness of the shell is d , which can vary over the element. The local coordinates ξ^1 and ξ^2 follow the midsurface, and ξ^3 is the coordinate in the normal direction. The normal coordinate has a value of $-d/2$ on the bottom side of the element, and $+d/2$ on the top side.

The position of the deformed midsurface is $\mathbf{r} + \mathbf{u}$, and the normal after deformation is $\mathbf{n} + \mathbf{a}$. To keep the normal a unit vector requires that

$$|\mathbf{n} + \mathbf{a}| = 1 \quad (5-1)$$

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: ξ^1 is the arc-length along the line on which the shell is modeled, and ξ^2 is a circumferential coordinate given by $\xi^2 = R\Theta$.

STRAINS

The in-plane Green-Lagrange strain in the local covariant components can then be written as

$$\varepsilon_{\alpha\beta} = \frac{1}{2} \left[\frac{\partial}{\partial \xi^\alpha} (\mathbf{r} + \mathbf{u} + \xi^3 (\mathbf{n} + \mathbf{a})) \cdot \frac{\partial}{\partial \xi^\beta} (\mathbf{r} + \mathbf{u} + \xi^3 (\mathbf{n} + \mathbf{a})) - \frac{\partial}{\partial \xi^\alpha} (\mathbf{r} + \xi^3 \mathbf{n}) \cdot \frac{\partial}{\partial \xi^\beta} (\mathbf{r} + \xi^3 \mathbf{n}) \right] = \gamma_{\alpha\beta} + \xi^3 \chi_{\alpha\beta} + (\xi^3)^2 \kappa_{\alpha\beta}$$

The indices α and β range from 1 to 2. The transverse shear strains in local covariant components are

$$\varepsilon_{3\alpha} = \varepsilon_{\alpha 3} = \frac{1}{2} \left[\frac{\partial}{\partial \xi^\alpha} (\mathbf{r} + \mathbf{u} + \xi^3 (\mathbf{n} + \mathbf{a})) \cdot (\mathbf{n} + \mathbf{a}) - \frac{\partial}{\partial \xi^\alpha} (\mathbf{r} + \xi^3 \mathbf{n}) \cdot \mathbf{n} \right] = \zeta_\alpha + \xi^3 \omega_\alpha$$

The constitutive relation for the shell elements is a plane stress assumption, as is customary in shell theory. The strain component in the normal direction ε_{33} is thus irrelevant. The different parts of the strain tensors above can be written out as

$$\begin{aligned} \gamma_{\alpha\beta} &= \frac{1}{2} \left[\frac{\partial \mathbf{u}}{\partial \xi^\alpha} \frac{\partial \mathbf{r}}{\partial \xi^\beta} + \frac{\partial \mathbf{r}}{\partial \xi^\alpha} \frac{\partial \mathbf{u}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha} \frac{\partial \mathbf{u}}{\partial \xi^\beta} \right] \\ \chi_{\alpha\beta} &= \frac{1}{2} \left[\frac{\partial \mathbf{r}}{\partial \xi^\alpha} \frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha} \frac{\partial \mathbf{r}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha} \frac{\partial \mathbf{n}}{\partial \xi^\beta} + \frac{\partial \mathbf{n}}{\partial \xi^\alpha} \frac{\partial \mathbf{u}}{\partial \xi^\beta} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha} \frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha} \frac{\partial \mathbf{u}}{\partial \xi^\beta} \right] \\ \kappa_{\alpha\beta} &= \frac{1}{2} \left[\frac{\partial \mathbf{a}}{\partial \xi^\alpha} \frac{\partial \mathbf{n}}{\partial \xi^\beta} + \frac{\partial \mathbf{n}}{\partial \xi^\alpha} \frac{\partial \mathbf{a}}{\partial \xi^\beta} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha} \frac{\partial \mathbf{a}}{\partial \xi^\beta} \right] \\ \zeta_\alpha &= \frac{1}{2} \left[\frac{\partial \mathbf{r}}{\partial \xi^\alpha} \cdot \mathbf{a} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha} \cdot \mathbf{n} + \frac{\partial \mathbf{u}}{\partial \xi^\alpha} \cdot \mathbf{a} \right] \end{aligned}$$

$$\omega_\alpha = \frac{1}{2} \left[\frac{\partial \mathbf{n}}{\partial \xi^\alpha} \cdot \mathbf{a} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha} \cdot \mathbf{n} + \frac{\partial \mathbf{a}}{\partial \xi^\alpha} \cdot \mathbf{a} \right]$$

In a geometrically linear analysis, the nonlinear terms (products between \mathbf{u} , \mathbf{a} , and their derivatives) disappear. In all study types, the contributions from the parts $\kappa_{\alpha\beta}$ and ω_α are ignored. They are small unless the element has an extremely high ratio between thickness and radius of curvature, in which case the errors from using shell theory are large anyway.



Note that ζ_α here is a strain component, not to be confused with the local coordinate in the normal direction or offset.

OFFSET

It is possible to model a shell with a midsurface that is not located at the meshed surface but at a certain offset from it. The offset is assumed to occur along the normal of the shell surface. In this case,

$$\mathbf{r}(\xi^1, \xi^2) = \mathbf{r}_R(\xi^1, \xi^2) + \zeta_0 \mathbf{n}(\xi^1, \xi^2)$$

where \mathbf{r}_R is the position of the meshed *reference surface* and ζ_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 in are not well defined in such situations, since the position of the midsurface may 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

$$\int_V f(\xi^1, \xi^2, \xi^3) dV = \int_{A-d/2}^{d/2} \int_A f_1(\xi^1, \xi^2) f_2(\xi^3) d\zeta dA = \int_A f_1(\xi^1, \xi^2) F_2(d) dA$$

All functions of ζ are assumed to be of the form ζ^n . Odd powers will integrate to zero, so typically the cross 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 Ω_{0i} is the prescribed rotation around the axis \mathbf{t}_i .

In a geometrically nonlinear analysis, it is not possible to prescribe individual elements of the rotation vector. If only one or two components have been specified, the remaining components are set to zero. The actual degrees of freedom are then computed as

$$\mathbf{a}_0 = (\mathbf{R}(\Omega_0) - \mathbf{I})\mathbf{n}$$

where \mathbf{R} is a standard rotation matrix, representing the finite rotation about the given rotation vector.

Initial velocities are always given using an angular velocity vector ω as

$$\dot{\mathbf{a}} = \omega \times (\mathbf{n} + \mathbf{a}_0)$$

SYMMETRY AND ANTISYMMETRY BOUNDARY CONDITIONS

It is possible to prescribe symmetry and antisymmetry boundary conditions. As a default, they are expressed in a shell local coordinate system. If applied to a boundary, the normal to the shell is assumed to be the normal to the symmetry or antisymmetry plane. The conditions are

$$\mathbf{u} \cdot \mathbf{n} = 0$$

$$\mathbf{a} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

for the antisymmetry case. Here \mathbf{t}_1 and \mathbf{t}_2 are two perpendicular directions in the plane of the shell.

When applied to an edge, there is a local coordinate system where \mathbf{t}_1 is the tangent to the edge, and \mathbf{t}_2 is perpendicular and in the plane of the shell. The assumption is then that \mathbf{t}_2 is the normal to the symmetry or antisymmetry plane. The constraints are

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the antisymmetry case.

When symmetry or antisymmetry conditions are specified in a general coordinate system with axis directions \mathbf{e}_i , $i = 1, 2, 3$ with \mathbf{e}_1 as the normal to the symmetry/antisymmetry plane the constraints are

$$\mathbf{u} \cdot \mathbf{e}_1 = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_2 \times \mathbf{n}) = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_3 \times \mathbf{n}) = 0$$

or the symmetry case and

$$\mathbf{u} \cdot \mathbf{e}_2 = 0$$

$$\mathbf{u} \cdot \mathbf{e}_3 = 0$$

$$\mathbf{a} \cdot (\mathbf{e}_1 \times \mathbf{n}) = 0$$

for the antisymmetry case. Using a general coordinate system sometimes leads to higher accuracy, since there is no element interpolation of the constraint directions involved.

Axisymmetric Models

In axisymmetric models, the only possible symmetry plane is the one having the Z -axis as normal. In this case, you can use the [Symmetry Plane](#) boundary condition. The imposed constraint is

$$w = 0$$

EXTERNAL LOADS

Contributions to the virtual work from the external load is of the form

$$\mathbf{u}_{\text{test}} \cdot \mathbf{F} + \mathbf{a}_{\text{test}} \cdot [\mathbf{M} \times (\mathbf{n} + \mathbf{a})]$$

where the forces (\mathbf{F}) and moments (\mathbf{M}) can be distributed over a boundary or an edge or concentrated in a point. The contribution from the normal vector displacement \mathbf{a} is only included in a geometrically nonlinear analysis. Loads are always referred to the midsurface of the element. In the special case of a follower load, defined by its pressure p , the force intensity is

$$\mathbf{F} = -p(\mathbf{n} + \mathbf{a})$$

For a follower load, the change in midsurface area is not taken into account, in order to be consistent with the assumption that thickness changes are ignored.

STRESS AND STRAIN CALCULATIONS

The strains calculated in the element are, as described above, the covariant tensor components. They have little significance for the user, and are internally transformed to a Cartesian coordinate system. This system can be global or element local. The stresses are computed by applying the constitutive law to the thus computed strain tensor.

Each part of the covariant strain ($\gamma_{\alpha\beta}$, $\chi_{\alpha\beta}$, ζ_{α}) is transformed separately. They correspond to membrane, bending, and shear action, respectively, and it is thus possible to separate the stresses from each of these actions. The membrane stress is defined as

$$\sigma_m = D[\gamma - \gamma_i - \alpha(T_m - T_{\text{ref}}) - \beta_h(c_m - c_{m0,\text{ref}})] + \frac{N_i}{d}$$

where D is the plane stress constitutive matrix, N_i are the initial membrane forces, and γ_i the initial membrane strains. The influence of thermal strains is included through the midsurface temperature T_m , and the hygroscopic swelling through the midsurface

moisture concentration, c_m . The membrane stress can be considered as the stress at the mid-surface, 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 χ_i is the initial value of the bending part of the strain tensor (actually: the curvature), and M_i are the initial bending and twisting moments. ΔT is the temperature difference between the top and bottom surface of the shell, and Δc_{mo} is the difference in moisture concentration between the top and bottom. The bending stress is the stress at the top surface of the shell if no membrane stress is present.

The average transverse shear stress is defined as

$$\sigma_s = \frac{5}{6} \cdot 2G(\zeta - \zeta_i) + \frac{Q_i}{d}$$

where G represent the transverse shear moduli, ζ_i is the initial average shear strain, and Q_i are the initial transverse shear forces. The correction factor $5/6$ ensures that the stresses are averaged so that they correspond to the ratio between shear force and thickness. The corresponding strains ζ and ζ_i are averaged in an energy sense.

The actual in-plane stress at a certain level in the element is then

$$\sigma = \sigma_m + z\sigma_b$$

where z is a parameter ranging from -1 (bottom surface) to $+1$ (top surface). The computation of the shear stress at a certain level in the element uses the following analytical parabolic stress distribution:

$$\frac{3\sigma_s(1-z^2)}{2}$$

The shell section forces (membrane forces, bending moments, and shear forces) are computed from the stresses as

$$N = d\sigma_m$$

$$M = \frac{d^2}{6}\sigma_b$$

$$Q = d\sigma_s$$

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 and 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.

The definition of the local shell surface coordinate system is as follows:

1. The local z direction \mathbf{e}_{z1} is the positive 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 X - Y 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 a 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* effective stresses. It is possible to evaluate the stress and strain tensor components and effective stresses at an arbitrary distance from the midsurface. The parameter z_{shell} (variable name `shell.z`) is found in, for example, the **Parameters** table of the **Settings** window of a surface plot. Its 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 Data Set

The [Shell](#) data set 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 data set with appropriate settings is generated together with the default plots, and some of the defaults 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 data set. 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 [Shell](#) data set. The Shell data set is only available in 3D, so it cannot be used by 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 like thermal expansion. You achieve 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:

$$\begin{aligned}
 m &= \int_V \rho dV = \int_A \rho d \, dA \\
 \mathbf{X}_M &= \frac{1}{m} \int_V \rho \mathbf{X} dV = \frac{1}{m} \int_V \rho (\mathbf{r} + \zeta \mathbf{n}) dV = \frac{1}{m} \int_A \rho d \mathbf{r} dA \\
 I &= \int_V (((\mathbf{X} - \mathbf{X}_M)^T \cdot (\mathbf{X} - \mathbf{X}_M)) E_3 - (\mathbf{X} - \mathbf{X}_M) \cdot (\mathbf{X} - \mathbf{X}_M)^T) \rho dV = \\
 &\int_A (((\mathbf{r} - \mathbf{X}_M)^T \cdot (\mathbf{r} - \mathbf{X}_M)) E_3 - (\mathbf{r} - \mathbf{X}_M) \cdot (\mathbf{r} - \mathbf{X}_M)^T) \rho d \, dA + \\
 &\int_A (E_3 - \mathbf{n} \mathbf{n}^T) \rho \frac{d^3}{12} dA
 \end{aligned}$$

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 taken into account.

CONNECTION BETWEEN SHELLS AND SOLIDS

This section describes the theory and assumptions behind the multiphysics couplings [Solid-Shell 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.

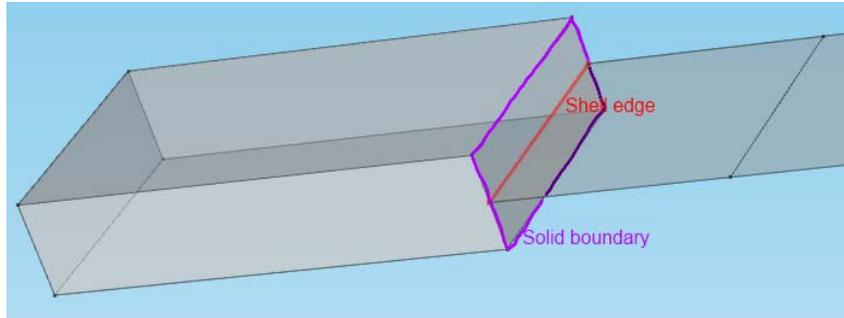


Figure 5-2: An example of a shell extending perpendicular to the solid boundary.

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:

$$\begin{aligned} u'_{sld}(y',z') &= u'_{sh}(y',0) + z'\mathbf{a}'_x(y',0) \\ v'_{sld}(y',z') &= v'_{sh}(y',0) + z'\mathbf{a}'_y(y',0) \end{aligned}$$

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

$$\begin{cases} \sigma_{x'} = A_1 + B_1 z' \\ \sigma_{y'} = A_2 + B_2 z' \\ \sigma_{z'} = 0 \\ \tau_{x'y'} = B_3 z' \\ \tau_{x'z'} = C_1(1 - z'^2) \\ \tau_{y'z'} = C_2(1 - z'^2) \end{cases}$$

where the relative thickness coordinate $z = \frac{2z'}{d}$ has been introduced.

Assuming that Hooke's law and plane stress conditions are valid, the transverse direct strain can be computed as

$$\varepsilon_{z'} = -\frac{\nu}{E}(\sigma_{x'} + \sigma_{y'}) = K_1(x', y') + 2K_2(x', y')z'$$

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

$$\begin{aligned} u'_{\text{slid}}(y',z') &= u'_{\text{sh}}(y',0) + z'a'_x(y',0) + q_u(y',0)(z^3 - k_u z)d \\ v'_{\text{slid}}(y',z') &= v'_{\text{sh}}(y',0) + z'a'_y(y',0) \\ w'_{\text{slid}}(y',z') &= w'_{\text{sh}}(y',0) + q_{w1}(y',0)zd + q_{w2}(y',0)(z^2 - k_{w2})d \end{aligned}$$

Here q_u , q_{w1} , and q_{w2} are the new dependent variables defined on the shell edge. They are dimensionless, due to the multiplication with the shell thickness, d . The constants $k_u = 3/5$ and $k_{w2} = 1/5$ are explained below. The variable q_{w1} is proportional to the membrane axial strain, the variable q_{w2} is proportional to the bending strain, and the variable q_u is proportional to the transverse shear strain. Since these variable are directly related to strains, the shape function order used is one order lower than for the displacements.

If no extra equations defining q_u , q_{w1} , and q_{w2} are introduced, these variables try to adapt to proper values through the reaction forces on the solid. The reaction force for u' is the traction $\sigma_{x'}$ and the reaction force for w' is the traction $\tau_{x'z'}$. When taking the variation of the new dependent variables, these enforce the following constraints:

$$\begin{aligned}
& \frac{d}{2} \\
& test(q_u) \int (z^3 - k_u z) \sigma_x dz' = 0 \\
& \frac{-d}{2} \\
& \frac{d}{2} \\
& test(q_{w1}) \int z \tau_{x'z'} dz' = 0 \\
& \frac{-d}{2} \\
& \frac{d}{2} \\
& test(q_{w2}) \int (z^2 - k_{w2}) \tau_{x'z'} dz' = 0 \\
& \frac{-d}{2}
\end{aligned}$$

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.

$$z' = (\mathbf{x}_{\text{sld}} - \mathbf{x}_{\text{sh}}) \cdot \mathbf{n}_{\text{sh}}$$

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 Φ 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

$$\Phi = [\mathbf{s} \mathbf{t}_1 \times \mathbf{N}_{\text{sh}} \quad \mathbf{s} \mathbf{t}_1 \quad \mathbf{N}_{\text{sh}}]$$

where \mathbf{N} is the undeformed shell normal (`shell.n`) and \mathbf{t}_1 is the shell edge tangent (`shell.t1`). 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 ζ is a distance from the solid to the shell. The right hand side is mapped from the shell to the solid using an extrusion operator. The default is that ζ is half the shell thickness, but you can also use the geometrical distance between the boundaries, or a user-defined distance.

CONNECTION BETWEEN SHELLS AND BEAMS

When connecting elements from the Shell interface with elements from the Beam interface, the following must be noted:

- The rotational degrees of freedom have different definitions in the two interfaces. The Beam interface uses a rotation vector representation, whereas the Shell interface uses the normal vector displacements.
- In general, a shell element locally has three translational, but only two rotational, degrees of freedom. The rotation around the normal is not part of the shell theory and is constrained to zero. The corresponding rotational degree of freedom in a beam should not thus not be connected to the shell.

You can create the appropriate couplings by adding a [Solid-Shell 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 center line. 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 the those components of the beam 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 complications 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.



- The **Layered Linear Elastic Material** is only available in the Composite Materials Module.
 - The **Layered Linear Elastic Material** is only available in 3D and not in 2D axisymmetry.
-

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 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. 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.

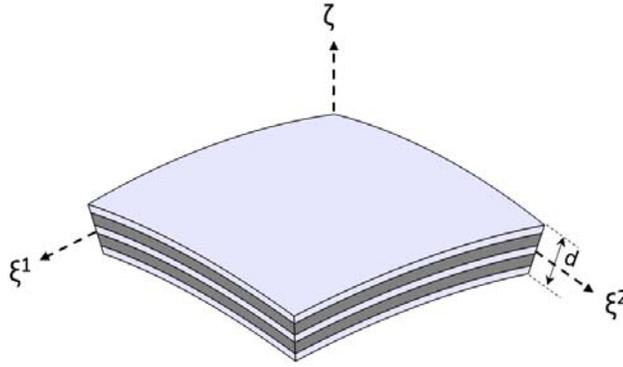


Figure 5-3: Geometry of a doubly curved composite laminate.

Figure 5-3 shows a doubly curved laminated shell with uniform total thickness, d . It is represented by orthogonal curvilinear coordinate system (ξ^1, ξ^2, ζ) . The geometry representation of a layered shell is same as a single layer shell as discussed in the [Geometry and Deformation](#) section.

A typical stacking sequence of a composite laminate having n layers is shown in [Figure 5-4](#). The thickness of each layer (d_k), as well as the fiber direction in each layer (θ_k) with respect to first principal direction (ξ^1) of the laminate are indicated. An

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

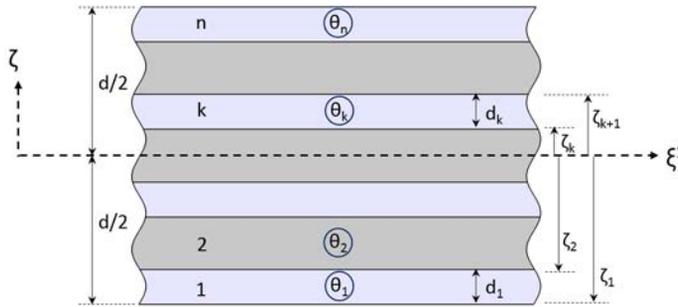


Figure 5-4: A typical stacking sequence of a composite laminate showing thickness and fiber orientation in each layer.

EQUIVALENT SINGLE LAYER (ESL) THEORY

Composite laminates are formed by stacking layers of different materials and/or having different fiber orientations. In general, composite laminates have a planar dimension which 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 non-zero, 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 (ϵ_{zz}) is non-zero but not part of the formulation.

FSDT extends the classical laminate theory and allows non-zero 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:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{bmatrix}^{(k)} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_6 \end{bmatrix}^{(k)}$$

where

- σ_{ij} are the stress components in the principal material directions of a lamina
- ϵ_{ij} are the strain components in the principal material directions of a lamina
- Q_{ij} are the elasticity matrix components in the principal material directions of a lamina

For an orthotropic lamina, the elasticity matrix components (Q_{ij}) can be defined in terms of the material constants ($E_1, E_2, \nu_{12}, \nu_{21}, G_{12}$):

$$Q_{11} = \frac{E_1}{(1 - \nu_{12}\nu_{21})}$$

$$Q_{22} = \frac{E_2}{(1 - \nu_{12}\nu_{21})}$$

$$Q_{12} = \frac{\nu_{12}E_2}{(1-\nu_{12}\nu_{21})} = \frac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})}$$

$$Q_{66} = G_{12}$$

A general composite laminate is made of several orthotropic layers, having principal material directions with different orientations with respect to the laminate coordinates (ξ^1, ξ^2, ζ) as shown in Figure 5-3. The constitutive equations for a lamina can be transformed from its material coordinate system to the laminate coordinate system, giving:

$$\begin{bmatrix} \sigma_{\xi^1\xi^1} \\ \sigma_{\xi^2\xi^2} \\ \sigma_{\xi^1\xi^2} \end{bmatrix}^{(k)} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^1\xi^1} \\ \varepsilon_{\xi^2\xi^2} \\ \varepsilon_{\xi^1\xi^2} \end{bmatrix}$$

where

- σ_{ij} are the stress components in the laminate coordinate system
- ε_{ij} are the strain components in the laminate coordinate system
- \bar{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 (θ) with the first in-plane direction (ξ^1) of the laminate coordinate system. Hence the transformation matrix can be defined as:

$$T = \begin{bmatrix} c^2 & s^2 & 2cs \\ s^2 & c^2 & -2cs \\ -sc & sc & c^2 - s^2 \end{bmatrix}$$

where $c = \cos\theta$ and $s = \sin\theta$.

Out-of-plane Constitutive Equations (Shear Equations)

The linear constitutive relation for the transverse shear in a lamina k can be written as:

$$\begin{bmatrix} \sigma_4 \\ \sigma_5 \end{bmatrix}^{(k)} = \begin{bmatrix} Q_{44} & 0 \\ 0 & Q_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}^{(k)}$$

where

- σ_{ij} are the transverse shear stress components in the principal material directions of a lamina
- ε_{ij} are the transverse shear strain components in the principal material directions of a lamina
- Q_{ij} are the elasticity matrix components in the principal material directions of a lamina

For an orthotropic lamina, the elasticity matrix components (Q_{ij}) can be defined in terms of the two shear moduli (G_{13}, G_{23}):

$$Q_{44} = G_{23}$$

$$Q_{55} = G_{13}$$

The constitutive equations of a lamina can be transformed from its material coordinate system to the laminate coordinate system and can be written as:

$$\begin{bmatrix} \sigma_{\xi^2\xi^3} \\ \sigma_{\xi^1\xi^3} \end{bmatrix}^{(k)} = \begin{bmatrix} \bar{Q}_{44} & \bar{Q}_{45} \\ \bar{Q}_{45} & \bar{Q}_{55} \end{bmatrix}^{(k)} \begin{bmatrix} \varepsilon_{\xi^2\xi^3} \\ \varepsilon_{\xi^1\xi^3} \end{bmatrix}$$

where

- σ_{ij} are the transverse shear stress components in the laminate coordinate system
- ε_{ij} are the transverse shear strain components in the laminate coordinate system
- \bar{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.

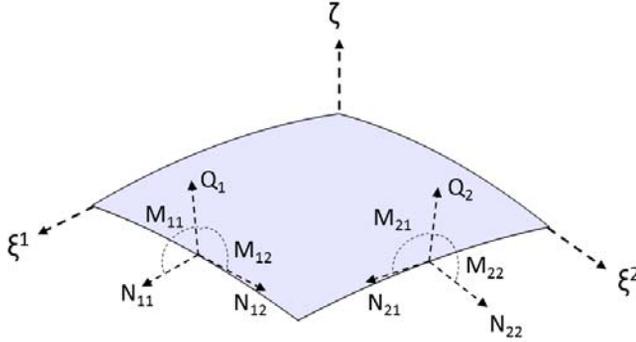


Figure 5-5: An equivalent layer of a composite laminate having n layers. The resultant membrane forces (N_{ij}), bending moments (M_{ij}), and transverse shear forces (Q_i) are shown.

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
- σ_{ij} are the stress components in the laminate coordinate system
- K_s is the shear correction factor
- R_1 and R_2 are the principal radii of curvature of the equivalent shell (actually this implies that the laminate system is aligned with the principal curvatures, but that is of no consequence, due to the approximations made below).

For shallow shells, the following approximation can be used:

$$\left(1 + \frac{\zeta}{R_1}\right) \approx 1$$

$$\left(1 + \frac{\zeta}{R_2}\right) \approx 1$$

This leads to

$$N_{12} = N_{21}$$

$$M_{12} = M_{21}$$

Hence the expressions for the resultant membrane force, bending moment, and transverse shear force reduce to the following:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1 \xi^1}^{(k)} \\ \sigma_{\xi^2 \xi^2}^{(k)} \\ \sigma_{\xi^1 \xi^2}^{(k)} \end{bmatrix} d\zeta$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^1 \xi^1}^{(k)} \\ \sigma_{\xi^2 \xi^2}^{(k)} \\ \sigma_{\xi^1 \xi^2}^{(k)} \end{bmatrix} \zeta d\zeta$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} \begin{bmatrix} \sigma_{\xi^2 \xi^3}^{(k)} \\ \sigma_{\xi^1 \xi^3}^{(k)} \end{bmatrix} d\zeta$$

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:

$$\begin{bmatrix} N_{11} \\ N_{22} \\ N_{12} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(0)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(0)} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(1)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(1)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} M_{11} \\ M_{22} \\ M_{12} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(0)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(0)} \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^1 \zeta^1}^{(1)} \\ \varepsilon_{\zeta^2 \zeta^2}^{(1)} \\ \varepsilon_{\zeta^1 \zeta^2}^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} Q_2 \\ Q_1 \end{bmatrix} = K_s \begin{bmatrix} A_{44} & A_{45} \\ A_{45} & A_{55} \end{bmatrix} \begin{bmatrix} \varepsilon_{\zeta^2 \zeta^3}^{(0)} \\ \varepsilon_{\zeta^1 \zeta^3}^{(0)} \end{bmatrix}$$

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:

$$\begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{bmatrix} \{\varepsilon^{(0)}\} \\ \{\varepsilon^{(1)}\} \end{bmatrix}$$

The midplane strains can be written using resultant forces and flexibility matrices as:

$$\begin{bmatrix} \{\varepsilon^{(0)}\} \\ \{\varepsilon^{(1)}\} \end{bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix}^{-1} \begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix}$$

$$\begin{bmatrix} \{\varepsilon^{(0)}\} \\ \{\varepsilon^{(1)}\} \end{bmatrix} = \begin{bmatrix} [a] & [b] \\ [b] & [d] \end{bmatrix} \begin{bmatrix} \{N\} \\ \{M\} \end{bmatrix}$$

where

- $[a]$ is the extensional flexibility matrix
- $[b]$ is the bending-extensional coupling flexibility matrix
- $[d]$ is the bending flexibility matrix

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 non-linear equations, it is assumed that there is no cross coupling between transverse shear stresses and transverse shear strains, i.e.

$$\bar{Q}_{45} = \bar{Q}_{54} = 0 .$$

- 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, transverse shear stress 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 zero 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, 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:

$$\sigma_i = G_i \cdot (E\zeta^2 + F\zeta + K_i) \cdot \varepsilon_{avg}$$

For n number of layers in a laminate, there are $n+2$ variables (unknowns) and thus $n+2$ equations are needed to solve them. The first $n+1$ equations are the form of shear stress continuity can be written as:

$$\sigma_1^{(bot)} = 0$$

$$\sigma_i^{(top)} = \sigma_{i+1}^{(bot)}$$

$$\sigma_n^{(top)} = 0$$

Here, the subscript indicates the layer index, and the superscript indicates the interface position (top or bottom) of a particular layer.

The missing equation can be expressed in the form of the average shear stress:

$$\sum_{k=1}^n \int_{\zeta^k}^{\zeta^{k+1}} \sigma_i d\zeta = \sum_{k=1}^n G_i \varepsilon_{avg} \zeta_i$$

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:

$$U_s = \frac{1}{2} \iint_A \int_{\zeta} \sigma \gamma d\zeta dA = \frac{1}{2} \iint_A \int_{\zeta} 4f(\zeta)^2 G(\zeta) \varepsilon_{avg}^2 d\zeta dA$$

where the shear strain distribution is defined as:

$$f(\zeta) = E\zeta^2 + F\zeta + K_i$$

The transverse shear strain energy, corresponding to (ε_{13}) , based on FSDT theory is written as:

$$U_s = \frac{1}{2K_{13}} \int_A 2Q_1 \varepsilon_{avg} dA = \frac{1}{2K_{13}} \int_A 4A_{55} \varepsilon_{avg}^2 dA$$

The shear correction factor can be obtained by equating the two energy equations:

$$K_{13} = \frac{\int_A A_{55} dA}{\iint_A \int_{\zeta} f(\zeta)^2 G_{13}(\zeta) d\zeta dA}$$

Similarly the shear correction factor corresponding to (ε_{23}) is:

$$K_{23} = \frac{\int_A A_{44} dA}{\iint_A \int_{\zeta} f(\zeta)^2 G_{23}(\zeta) d\zeta dA}$$

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 effective 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 effective 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 though 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 data set

Layered Material Data Set

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

Using this data set, 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 data set 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 data set for better visualization.

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 3D boundaries. 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 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 **Quasi-static**. Use **Quasi-static** 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 **Quasi-static** 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}_{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.

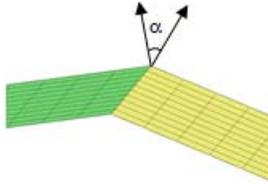
FOLD-LINE LIMIT ANGLE



This section is available for the Shell interface only. Also see [The MITC Shell Formulation](#).

The fold-line limit angle α is the smallest angle between the normals of two boundaries that makes their intersection to be treated as a fold line. The normal to the shell is discontinuous along a fold-line. Enter a value or expression in the α field. The default

value is 0.05 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** button () and select **Advanced Physics Options**. 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.

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.

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.

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 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
- 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
- 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:

- Applied Force (Rigid Connector)
- Applied Moment (Rigid Connector)
- Free
- Mass and Moment of Inertia (Rigid Connector)
- Gravity¹
- Point Load (on Axis)
- Rigid Domain
- Rotating Frame¹

¹ This is selected from the **Face and Volume Loads** submenu for this interface.



Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics Reference Manual*



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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.

- For **Physical offset**, enter a value or expression in the z_{offset} field for the actual distance from the meshed boundary to the shell midsurface.

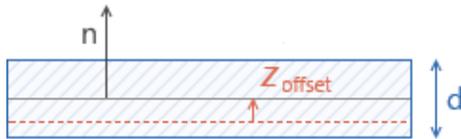


Figure 5-7: Meshed boundary indicated in red. The vector \mathbf{n} indicates the positive orientation of the shell normal.



The offset settings are not available for the Plate interface.

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.



For theory, see [Offset](#).

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 , ν_{yz} , or ν_{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](#).

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, or a laminated composite 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.



For a general description about layered materials, see [Layered Materials](#) in the documentation for the Composite Materials Module.

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](#)
- [Safety](#)

The **Layered Linear Elastic Material** node is only available with the Composite Materials Module (see <http://www.comsol.com/products/specifications/>).



The **Layered Linear Elastic Material** node is only available in 3D and not in 2D axisymmetry.

LAYER SELECTION

In the **Layer** drop-down, select a **Layered Material Link** or **Layered Material Stack** defined under **Materials** in the same component as the Shell interface.

At this node, it is not possible to select only few layers from a layered material as all the layers of a layered material are selected by default once a **Layered Material Link** or **Layered Material Stack** is chosen.



For a general description of this section, see [Layer Selections](#) in the documentation for the Composite Materials Module.

BOUNDARY SELECTION

The boundary selection on this node is similar to the **Linear Elastic Material** node however only those boundaries are allowed to select 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. Note that:

- For **Orthotropic** no values for E_z , ν_{yz} , or ν_{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.

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.

SHEAR CORRECTION FACTOR

In this section there is a drop-down input to define 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** button () and select **Advanced Physics Options**.

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 and 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 additional coordinate system that the model includes. The coordinate system given 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 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

Plate>Linear Elastic Material>Shell Local System

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes>Shell Local System

Thermal Expansion (for materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature. The thermal strain depends on the coefficient of thermal expansion (CTE) α , the temperature T , and the strain-free reference temperature T_{ref} as

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

It is possible to model bending due to a temperature gradient in the thickness direction of the shell. The temperature is then assumed to vary linearly through the thickness.

LAYER SELECTION

This section is only present when this node is added under **Layered Linear Elastic Material** node. In this section, select the layers in which thermal expansion needs to be modeled.



For a general description of this section, see [Layer Selections](#) in the documentation for the Composite Materials Module.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.

From the **Temperature** T list, select an existing temperature variable from a heat transfer interface (for example, **Temperature (htsh/sol1)**), if any temperature variables exist. For **User defined** enter a value or expression for the temperature. This is the mid-surface temperature of the shell, controlling the membrane part of the thermal expansion. If

needed, you can add a through-thickness temperature gradient in the **Thermal Bending** section.



See also [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain.

From the **Coefficient of thermal expansion α** list, select **From material** to use the coefficient of thermal expansion from the material, or **User defined** to enter a value or expression for α . Select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of thermal expansion tensor α . 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

Enter the **Temperature difference in thickness direction ΔT_z** , which affects the thermal bending. This is the difference between the temperatures at the top and bottom surfaces.



This section is not available when this node is added under **Layered Linear Elastic Material** node.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Thermal Expansion

Shell>Layered Linear Elastic Material>Thermal Expansion

Plate>Linear Elastic Material>Thermal Expansion

Ribbon

Physics tab with **Linear Elastic Material** or **Layered 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

$$\epsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the thickness direction of the shell. The concentration is then assumed to vary linearly through the thickness.

LAYER SELECTION

This section is only present when this node is added under **Layered Linear Elastic Material** node. In this section, select the layers in which hygroscopic swelling needs to be modeled.



For a general description of this section, see [Layer Selections](#) in the documentation for the Composite Materials Module.

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.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** (the default) or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref} . This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as **Concentration type**, also enter the **Molar mass** of the fluid, M_m . The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_h uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of hygroscopic swelling tensor β_h . The default value for the **User defined** case is $1.5e-4 \text{ m}^3/\text{kg}$. 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

Enter the **Concentration difference in thickness direction** Δc_{gy} , which affects the hygroscopic bending. This is the difference between the concentrations at the top and bottoms surfaces.



This section is not available when this node is added under **Layered Linear Elastic Material** node.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Hygroscopic Swelling

Shell>Layered Linear Elastic Material>Hygroscopic Swelling

Plate>Linear Elastic Material>Hygroscopic Swelling

Ribbon

Physics tab with **Linear Elastic Material** or **Layered 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](#) or [Layered 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.

LAYER SELECTION

This section is only present when this node is added under **Layered Linear Elastic Material** node. In this section, select the layers in which initial stress and strain needs to be modeled.



For a general description of this section, see [Layer Selections](#) in the documentation for the Composite Materials Module.

COORDINATE SYSTEM SELECTION

The given initial stresses and strains are interpreted in this system.

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](#).

INITIAL STRESS

Specify the initial stress as the **Initial in-plane force**, the **Initial moment**, and the **Initial out-of-plane shear force**. Enter values or expressions in the applicable fields for the:

- **Initial in-plane force** N_0
- **Initial moment** M_0
- **Initial out-of-plane shear force** Q_0

INITIAL STRAIN

Specify the initial strain as the **Initial membrane strain**, the **Initial bending strain**, and the **Initial transverse shear strain**. Enter values or expressions in the applicable fields for the:

- **Initial membrane strain** γ_0

- **Initial bending strain** χ_0
- **Initial transverse shear strain** ζ_0



- For definitions of the generalized strains, see [Theory for 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

Plate>Linear Elastic Material>Initial Stress and Strain

Ribbon

Physics tab with **Linear Elastic Material** or **Layered Linear Elastic Material** 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.



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.

LAYER SELECTION

This section is only present when this node is added under **Layered Linear Elastic Material** node. In this section, select the layers in which initial stress and strain needs to be modeled.



For a general description of this section, see [Layer Selections](#) in the documentation for the Composite Materials Module.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate system that the model includes. The user-defined external stresses and strains are interpreted in this system. The coordinate system given is projected onto the shell surface as described in [Local Coordinate Systems](#).

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor** or **In-plane force**.

- 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** drop-down 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_{ext} , the **Moment** M_{ext} , and the **Out-of-plane shear force** Q_{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

Plate>Linear Elastic Material>External Stress

Ribbon

Physics tab with **Linear Elastic Material** or **Layered Linear Elastic 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**, and **Layered Linear Elastic Material**. Four different variables describing the failure risk are defined, as described in [Table 5-1](#).

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 5-1: 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.



All other settings for the **Safety** node are described in the documentation for [Safety](#) in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Safety

Shell>Layered Linear Elastic Material>Safety

Ribbon

Physics tab with **Linear Elastic Material** or **Layered Linear Elastic Material** node selected in the model builder tree:

Attributes>External 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**, and **Layered Linear Elastic Material**.



All other settings for the **Damping** node are described in the documentation for [Damping](#) in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Damping

Shell>Layered Linear Elastic Material>Damping

Ribbon

Physics tab with **Linear Elastic Material** or **Layered Linear Elastic Material** node selected in the model tree:

Attributes>Damping

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** button () and select **Advanced Physics Options**.



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 θ 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, θ .

- **Normal vector** to describe the rotational degrees of freedom as a prescribed normal vector. Enter the components of the **Prescribed normal vector** N_0 .

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.



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** drop-down 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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of 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** α . 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 α . 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_{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

Ribbon

Physics tab with **Fixed Constraint** or **Prescribed Displacement** 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** can not 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** can not 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** button () and select **Advanced Physics Options**.



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** button () and select **Advanced Physics Options**.



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** button () and select **Advanced Physics Options**.



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)

Specify the coordinate system to use for specifying a symmetry edge. From the **Coordinate system** list select from:

- **Local edge system** (the default).



For details about the definition of local edge systems, see [Local edge system](#).

- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate system.



This section is available only for edges in the Shell interface.

FACE DEFINING THE LOCAL ORIENTATIONS

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**.



This section is available only for edges in the Shell interface.

SYMMETRY

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.



This section is available only for edges in the Shell interface.

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 drop-down 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** button () and select **Advanced Physics Options**.



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

In an axially symmetric problem, the only possible symmetry plane is has the Z -axis as normal. Use the **Symmetry Plane** node to impose such a symmetry condition.



This node is only available in the 2D axisymmetric Shell interface.

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 drop-down 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** button () and select **Advanced Physics Options**.



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).



For details about the definition of local edge systems, see [Local edge system](#).

- **Global coordinate system** (the standard global coordinate system).
- Any additional user-defined coordinate 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** button () and select **Advanced Physics Options**.

	<p>In the <i>COMSOL Multiphysics Reference Manual</i>:</p> <ul style="list-style-type: none">• Constraint Reaction Terms• Weak Constraints• Constraint Settings
	<p>Symmetry and Antisymmetry Boundary Conditions</p>
	<p>You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling 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**.

LOAD TYPE	VARIABLE	SI UNIT	GEOMETRIC ENTITY LEVEL	SPACE DIMENSION (COMPONENTS)
Force per unit area	F_A	N/m^2	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 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** (Shell only)



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) .

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 ²	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.

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 ²	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 5-2](#) 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 **Anisotropic**. For **Isotropic** the same spring constant is used in all the diagonal elements of the spring matrix.

TABLE 5-2: 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 5-2: 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** η_s which is used to multiply all values of the spring matrix or spring force vector.
- For **Individual components** select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic**, then enter values or expressions in the table for the **Loss factor for spring** η_k or η_f based on space dimension. The loss factors act on the corresponding components of the spring matrix or spring force vector. If you select **Isotropic**, the effect is the same as when you select **Diagonal** and enter the same value for all diagonal elements.

VISCOUS DAMPING

Select the **Damping type** using [Table 5-3](#) 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 Anisotropic. For **Isotropic** the same viscous constant is used in all the diagonal elements of the damping matrix

TABLE 5-3: DAMPING TYPES FOR THE SPRING FOUNDATION FEATURE

DAMPING TYPE	VARIABLE	SI UNITS	GEOMETRIC ENTITY LEVEL
Damping constant per unit volume	d_V	$N \cdot s / (m \cdot m^3)$	domains (Plate), boundaries (Shell)
Damping constant per unit area	d_A	$N \cdot s / (m \cdot m^2)$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Total damping constant	d_{tot}	$N \cdot s / m$	domains (Plate), boundaries (Shell, Plate), edges (Shell)
Damping constant per unit length	d_L	$N \cdot s / (m \cdot m)$	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** α_{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* respectively. 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 respectively. 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, as long as the edges do not intersect at the longitudinal axis.

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 **Sector angle**. If **Automatic** is selected, the program attempts to find out how many full repetitions of the geometry there will be on a full revolution. If **User defined** is selected, enter a value for the sector angle θ_S .



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 θ_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 $u_i(\mathbf{x}_d) = -u_i(\mathbf{x}_s)$ respectively. 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** button () and select **Advanced Physics Options**.



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*.



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

Solid Connection



The **Solid Connection** node is obsolete, and has been superseded by the [Solid-Shell 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.

In 3D, 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** ζ . For **Offset vector** enter values for \mathbf{d}_0 in the table. The offset is the vector from the reference surface of the shell to the actual position of the beam. 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. The selected shell edges will move as a single rigid object, irrespective of whether they are geometrically adjacent or not.

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.

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** 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 **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: Edge** or **Center of Rotation: Point** subnode is added, depending on the setting of **Entity level**.

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_{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** Ω and an **Angle of rotation** ϕ . The axis of rotation is given in the selected coordinate system.



- 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>Connections>Rigid Connector

Ribbon

Physics tab with **Shell** selected:

Edges>Connections>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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

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** α . 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 α . 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_{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 the addition of 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

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

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** α . 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 α . 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_{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 (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

PHASE

Add the phase angle \mathbf{F}_{ph} for harmonic loads. Enter the phase for each component of the load in the corresponding fields.

MOMENT LOAD PHASE

Add the phase for the moment load \mathbf{M}_{ph} for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

LOCATION IN USER INTERFACE

Context Menus

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.

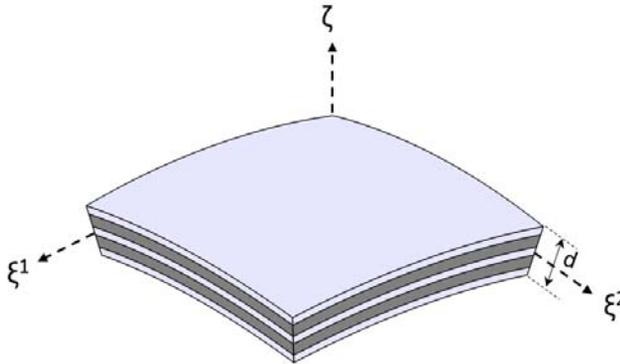


Figure 6-1: Geometry of a doubly curved laminated shell.

Figure 6-1 shows the uniform thickness doubly curved laminated shell having an orthogonal curvilinear coordinate system (ξ^1, ξ^2, ζ) and a total thickness (d).

A typical stacking sequence of a composite laminate having n layers is shown in **Figure 6-2**. The thickness of each layer (d_k), as well as the fiber direction in each layer

(θ_k) with respect to first principal direction (ξ^1) of the laminate are indicated. A counterclockwise rotation of the fiber direction with respect to (ζ) direction is considered as positive.

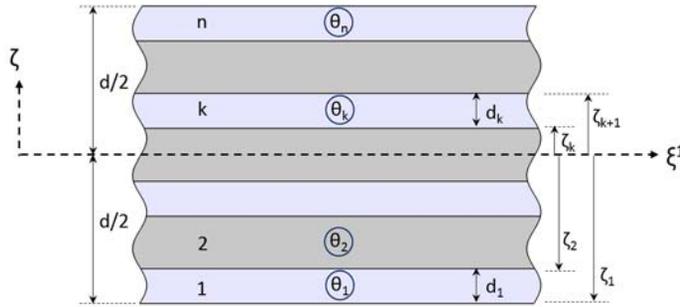


Figure 6-2: A typical stacking sequence of a composite laminate showing thickness and fiber orientation in each layer.



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](#)
- [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 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

- ∇_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

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla_T \mathbf{u} \cdot \left(\frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} \right)^{-1} + \frac{d\mathbf{u}}{dz} \mathbf{n}^T + I$$

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}_r} = I + \left(z + \frac{d}{2}(-1 + z_{off}) \right) \nabla_T \mathbf{n}$$

where

- \mathbf{X}_r are the reference surface coordinates
- z_{off} is the relative midplane offset
- d is the laminate thickness

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:

$$ASF = \left| \frac{\partial \mathbf{X}}{\partial \mathbf{X}_b} \right|$$

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:

$$\int_V f(\xi^1, \xi^2, \zeta) dV = \int_A \int_{\zeta} f_1(\xi^1, \xi^2) f_2(\zeta) d\zeta dA = \int \sum_{k=1}^n \int_{\zeta_k}^{\zeta_{k+1}} f_1(\xi^1, \xi^2) f_2(\zeta) dA$$

As discussed in the previous section, an area scale factor 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.

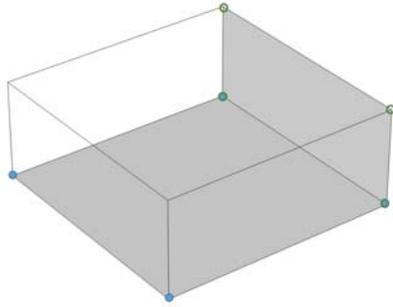
DISCRETIZATION

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)

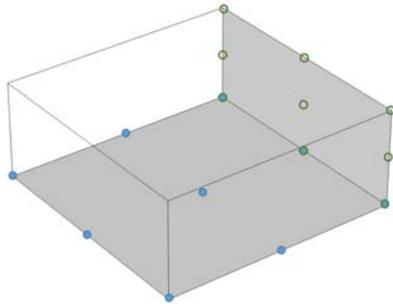
Based on the above two categories, 9 different elements are available:

- 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)

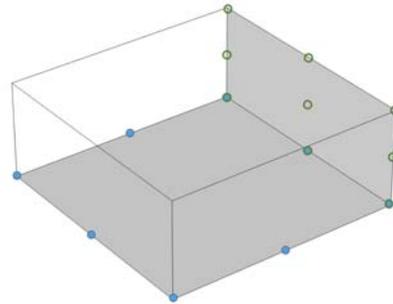


Linear (8 nodes)

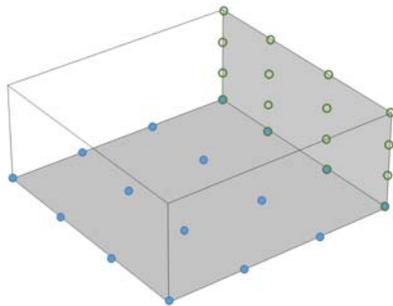
Figure 6-3: Element having linear shape order.



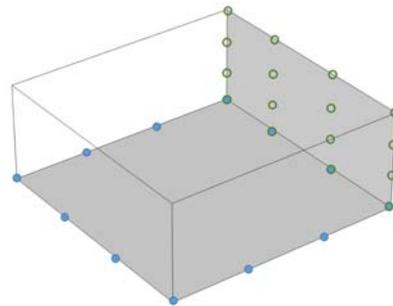
Quadratic Lagrange (27 nodes)



Quadratic Serendipity (24 nodes)

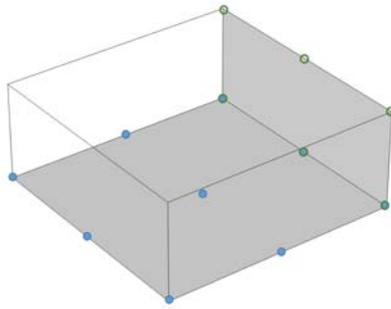


Cubic Lagrange (64 nodes)

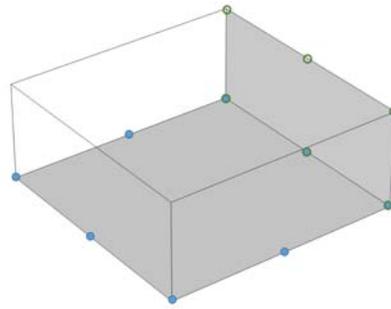


Cubic Serendipity (48 nodes)

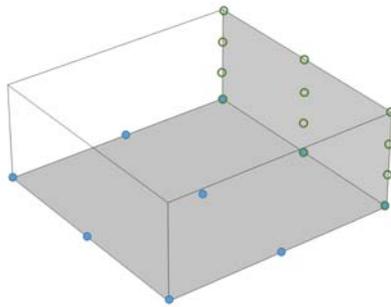
Figure 6-4: Elements having quadratic and cubic shape orders.



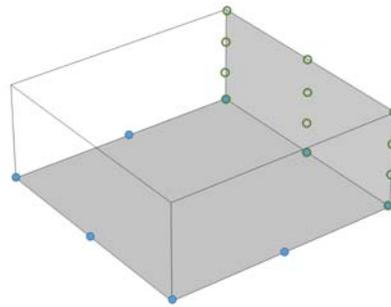
Quadratic-Linear Lagrange (18 nodes)



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.

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

$$\mathbf{n}_j \cdot \mathbf{n}_k = \mathbf{u}_{rj} \cdot \mathbf{u}_{rk}$$

$$\mathbf{u}_r = \left(z + \frac{d}{2}(-1 + z_{off}) \right) \mathbf{n} + \mathbf{u} - \mathbf{u}_b$$

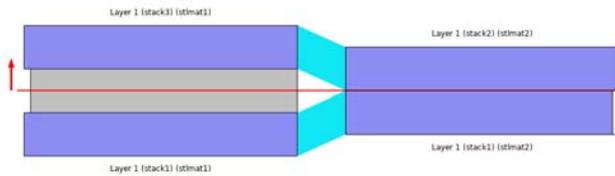
where \mathbf{u}_r is the displacement vector in the through thickness direction relative to the displacement vector at the reference surface location.

LAYERED SHELL CONTINUITY

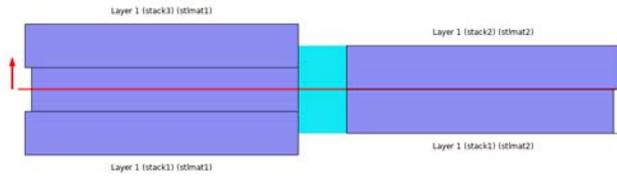
Different layered materials have their own virtual extra dimension geometry and that is why by default two layered materials sharing a common edge in the geometry do not have a continuous displacement field. Thus, when two layered shell surfaces having different layered materials meet side-by-side, a 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, some of the ways you can connect the two layered materials in the through thickness direction are as follows:

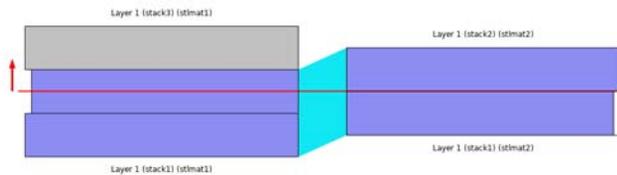
- Connecting the first and third layer of the left layered material to both the layers of the right layered material. This type of connection is typically used in a ply dropping scenario.



- Connecting the geometrically common area of both the layered materials in through thickness direction.



- Connecting the first two layers of 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.

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 d) and `1shell.xd_re1` (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.

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 effective 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 effective 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 though 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 data set

Layered Material Data Set

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

Using this data set, 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 data set 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 data set for better visualization.

The Layered Shell Interface

The **Layered Shell (Ishell)** 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 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 <http://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 `Ishell1`.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasi-static**. Use **Quasi-static** 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 **Quasi-static** is selected.

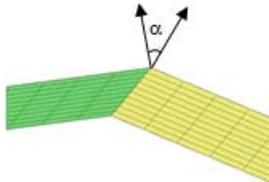
This is often the case when the time dependence exists only in some other physics, like a transient heat transfer problem causing thermal strains.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation** \mathbf{x}_{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 α is the smallest angle between the normals of two boundaries that makes their intersection to be treated as a fold line. The normal to the layered shell is discontinuous along a fold-line. Enter a value or expression in the α field. The default value is 0.05 radians (approximately 3°). The value must be larger than 0, and less than $\pi/2$, but angles larger than a few degrees are not usually meaningful.



See [Fold-Line Connection](#) 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 Layer Selection Section

Almost all physics nodes that you add in the Layered Shell interface have a section named **Layer Selection**. Selections there 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 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.



See [Physics Interface Selection](#) for the more details.

For a general description of layer selections, see [Layer 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):

- [Added Mass](#)
- [Added Mass, Interface](#)
- [Antisymmetry](#)
- [Attachment](#)
- [Body Load](#)
- [Boundary Load](#)
- [Continuity](#)
- [Creep](#)
- [Damping](#)
- [Edge Load](#)
- [External Stress](#)
- [External Strain](#)
- [Face Load](#)
- [Fixed Constraint](#)
- [Fixed Constraint, Interface](#)
- [Free](#)
- [Gravity](#)
- [Hygroscopic Swelling](#)
- [Initial Values](#)
- [Initial Stress and Strain](#)
- [Linear Elastic Material](#)
- [Line Load](#)
- [Phase](#)
- [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](#)
- [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 2-3](#) for links to common sections and [Table 2-4](#) 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.

LAYER SELECTION

Select the material layers to which the initial values are to be applied.

TABLE 6-1: LAYER SELECTIONS; INITIAL VALUES

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	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 Layer Selection Section](#).
- 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](#)
- [Creep](#)
- [Viscoplasticity](#)
- [Safety](#)

LAYER SELECTION

Select the layer or layers for which this material model is to be used.

TABLE 6-2: LAYER SELECTIONS; LINEAR ELASTIC MATERIAL

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	When a single layered material definition is 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 **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 Layer Selection Section](#).
- 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

Viscoelasticity

Use the **Viscoelasticity** subnode to add viscous stress contributions to an elastic material model.

LAYER SELECTION

Select the layer or layers for which this viscoelastic model is to be used. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

When 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 Layer Selection Section](#).
 - 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

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:
Attributes>Viscoelasticity

Creep

Use the **Creep** subnode to define the creep properties of the material model.

The Nonlinear Structural Material Module is required for this material model.

LAYER SELECTION

Select the layer or layers for which this creep model is to be used. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

When the creep model differs between layers, you will need to add several **Creep** nodes with different layer selections. 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 Layer Selection Section](#).
- 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.

LAYER SELECTION

Select the layer or layers for which this viscoplasticity model is to be used. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 with **From material**. 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 Layer Selection Section](#).
- 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.

You can provide a temperature with a through-thickness variation by explicitly or implicitly use expressions containing the extra dimension coordinate as described in [Using the Extra Dimension Coordinates](#).

LAYER SELECTION

Select the layer or layers for which this thermal strain contribution is to be added. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

If the **Input type** differs between layers, you will need to add several **Thermal Expansion** nodes with different layer selections. If only the material data values differ, you can use a single **Thermal Expansion** node with **From material**. The data for each layer is then received from global **Layered Material** nodes.

If the same layer is selected in two **Thermal Expansion** nodes being active on the same boundary, the second definition will override the previous.



- All other settings 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 Layer Selection Section](#).
 - 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](#).
-

Context Menus

Layered Shell>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

$$\varepsilon_{hs} = \beta_h M_m (c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, M_m is the molar mass, c_{mo} is the moisture concentration, and $c_{mo,ref}$ 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](#).

LAYER SELECTION

Select the layer or layers for which this swelling strain contribution is to be added. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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.



- 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 Layer Selection Section](#).
 - 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

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.

LAYER SELECTION

Select the layer or layers for which this stress or strain contribution is to be added. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 Layer Selection Section](#).
 - 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](#) 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.

LAYER SELECTION

Select the layer or layers for which this stress contribution is to be added. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 Layer Selection Section](#).
 - 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

Ribbon

Physics tab with **Linear Elastic 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](#) on a variety of formats.

LAYER SELECTION

Select the layer or layers for which this strain contribution is to be added. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 Layer Selection Section](#).
- 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

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model builder tree:

Attributes>External Strain

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.

LAYER SELECTION

Select the layer or layers for which the safety factor variables are to be defined. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 Layer Selection Section](#).
- 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>Safety

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model builder tree:

Attributes>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.

LAYER SELECTION

Select the layer or layers for which this damping model is to be used. The **Layer** selector inherits the value set in the parent **Linear Elastic Material** node. When it is set to **All layered materials**, you cannot access the individual layers.

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 Layer Selection Section](#).
- 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

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes>Damping

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.

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.

LAYER SELECTION

In the **Rigid Domain** node, you cannot access individual layers. Thus the selection in this section only acts as a filter for the boundary selection. All layers through the shell thickness act together as a single rigid body.

TABLE 6-3: LAYER SELECTIONS; RIGID DOMAIN

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available



- 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 Layer Selection Section](#).

LOCATION IN USER INTERFACE

Context Menus

Layered Shell>Material Models>Rigid Domain

Ribbon

Physics tab with **Layered Shell** selected:

Boundaries>Material Models>Rigid Domain

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.

LAYER SELECTION

In the **Free** node, you cannot access individual layers. Thus the selection in this section only acts as a filter for the boundary selection.

TABLE 6-4: LAYER SELECTIONS; FREE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available

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 Layer Selection Section](#).

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](#).

LAYER SELECTION

In the **Prescribed Displacement** node, you cannot access individual layers. Thus 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 6-5: LAYER SELECTIONS; PRESCRIBED DISPLACEMENT

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available
Edge	Default	Selectable	Not available
Point	Default	Selectable	Not available



- 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 Layer Selection Section](#).
- 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](#).

LAYER SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 6-6: LAYER SELECTIONS; PRESCRIBED DISPLACEMENT, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected
Edge	No	Selectable	All interfaces can be selected
Point	No	Selectable	All interfaces can be selected



- 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 Layer Selection Section](#).

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](#).

LAYER SELECTION

In the **Prescribed Velocity** node, you cannot access individual layers. Thus 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 6-7: LAYER SELECTIONS; PRESCRIBED VELOCITY

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available
Edge	Default	Selectable	Not available



- 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 Layer Selection Section](#).
- 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](#).

LAYER SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 6-8: LAYER SELECTIONS; PRESCRIBED VELOCITY, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected



- 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 Layer Selection Section](#).

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](#).

LAYER SELECTION

In the **Prescribed Acceleration** node, you cannot access individual layers. Thus 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 6-9: LAYER SELECTIONS; PRESCRIBED ACCELERATION

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available
Edge	Default	Selectable	Not available



- 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 Layer Selection Section](#).
- 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](#).

LAYER SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 6-10: LAYER SELECTIONS; PRESCRIBED ACCELERATION, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected



- 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 Layer Selection Section](#).

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 entities to zero. It is available for boundaries, edges, and points.

To constrain interfaces between the material layers, use [Fixed Constraint, Interface..](#)

LAYER SELECTION

In the **Fixed Constraint** node, you cannot access individual layers. Thus 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 6-11: LAYER SELECTIONS; FIXED CONSTRAINT

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available
Edge	Default	Selectable	Not available
Point	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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](#).

LAYER SELECTION

Select the interface or interfaces for which this constraint is to be added.

TABLE 6-12: LAYER SELECTIONS; FIXED CONSTRAINT, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected
Edge	No	Selectable	All interfaces can be selected
Point	No	Selectable	All interfaces 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 Layer Selection Section](#).

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

LAYER SELECTION

Select the layer or interfaces based on this node is added under which type of constraints feature. You can select individual layers or interfaces in this node which are allowed on the parent node.



All other settings for the **Thermal Expansion** node are described in the documentation for [Thermal Expansion \(for Constraints\)](#) in the Solid Mechanics interface.

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](#).

LAYER SELECTION

Select the material layers to which this constraint is to be applied.

In the **Roller** node, you cannot access individual layers. Thus 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 6-13: LAYER SELECTIONS; ROLLER

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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](#).

LAYER SELECTION

Select the interface or interfaces at which the constraint is to be applied.

TABLE 6-14: LAYER SELECTIONS; ROLLER, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces 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 Layer Selection Section](#).

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.

LAYER SELECTION

Select the material layers to which this constraint is to be applied.

In the **Symmetry** node, you cannot access individual layers. Thus 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 6-15: LAYER SELECTIONS; SYMMETRY

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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.

LAYER SELECTION

Select the material layers to which this constraint is to be applied.

In the **Antisymmetry** node, you cannot access individual layers. Thus 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 6-16: LAYER SELECTIONS; ANTISYMMETRY

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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.

LAYER SELECTION

In the **Rigid Motion Suppression** node, you cannot access individual layers. Thus the selection in this section only acts as a filter for the boundary selection.

TABLE 6-17: LAYER SELECTIONS; RIGID MOTION SUPPRESSION

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not possible



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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-18: LAYER SELECTIONS; BODY LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	When a single layered material definition is selected.



All other settings for the **Body Load** node are described in the documentation for [Body Load](#) in the Solid Mechanics interface.

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.

FORCE

Select a **Load type**—**Force per unit area**, **Total force**, or **Pressure**.

LOAD TYPE	VARIABLE	SI UNIT
Force per unit area	F_A	N/m^2
Total force	F_{tot}	N
Pressure	p	Pa



- 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.

LAYER SELECTION

Select the interface or interfaces at which the load is to be applied.

TABLE 6-19: LAYER SELECTIONS; FACE LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected



- For a general description of layer and interface selections, see [The Layer Selection Section](#).

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-20: LAYER SELECTIONS; GRAVITY

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-21: LAYER SELECTIONS; ROTATING FRAME

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Not available



- 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 Layer Selection Section](#).

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-22: LAYER SELECTIONS; BOUNDARY LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	When a single layered material definition is selected.



All other settings for the **Boundary Load** node are described in the documentation for [Boundary Load](#) in the Solid Mechanics interface.

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-23: LAYER SELECTIONS; EDGE LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Edge	No	Selectable	When a single layered material definition is selected.



All other settings for the **Edge Load** node are described in the documentation for [Edge Load](#) in the Solid Mechanics interface.

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-24: LAYER SELECTIONS; LINE LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL LAYERS
Point	Default	Selectable	When a single layered material definition is selected.



All other settings for the **Line Load** node are described in the documentation for [Edge Load](#) in the Solid Mechanics interface.

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.

LAYER SELECTION

Select the material layers to which this load is to be applied.

TABLE 6-25: LAYER SELECTIONS; POINT LOAD

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL DEFINITIONS	SELECTION OF INDIVIDUAL INTERFACES
Point	No	Selectable	When a single layered material definition is selected.



All other settings for the **Point Load** node are described in the documentation for [Point Load](#) in the Solid Mechanics interface.

LOCATION IN USER INTERFACE

Context Menus

Layered Shell>Points>Interfaces>Point Load

Ribbon

Physics tab with **Layered Shell** selected:

Points>Interfaces>Point Load

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response the physics interface splits the harmonic load into two parameters:

- The amplitude, F , which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

LAYER SELECTION

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 **TPoint 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.

By adding the [Predeformation](#) subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

LAYER SELECTION

Select the material layers to which the spring contribution is to be applied.

[

TABLE 6-26: LAYER SELECTIONS; SPRING FOUNDATION

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	When a single layered material definition is selected
Edge	Default	Selectable	When a single layered material definition is selected.
Point	Default	Selectable	When a single layered material definition is selected.

For a general description of this section, see [The Layer Selection Section](#).



All other settings for the **Spring Foundation** node are described in the documentation for [Spring Foundation](#) in the Solid Mechanics interface.

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.

By adding the [Predeformation](#) subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

LAYER SELECTION

Select the interfaces between the layers to which the spring contribution is to be applied.

TABLE 6-27: LAYER SELECTIONS; SPRING FOUNDATION, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	When a single layered material definition is selected
Edge	No	Selectable	When a single layered material definition is selected.
Point	No	Selectable	When a single layered material definition is selected.

For a general description of this section, see [The Layer Selection Section](#).



All other settings for the **Spring Foundation, Interface** node are described in the documentation for [Spring Foundation](#) in the Solid Mechanics interface.

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.

By adding the [Predeformation](#) subnode, you can prescribe that the spring force is zero at a nonzero spring extension.

LAYER SELECTION

TABLE 6-28: LAYER SELECTIONS; THIN ELASTIC LAYER

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	When a single layered material definition is selected.

For a general description of this section, see [The Layer Selection Section](#).



All other settings for the **Thin Elastic Layer** node are described in the documentation for [Thin Elastic Layer](#) in the Solid Mechanics interface.

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** 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.

LAYER SELECTION

[

TABLE 6-29: LAYER SELECTIONS; THIN ELASTIC LAYER, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	Only interior interfaces when a single layered material definition is selected.

For a general description of this section, see [The Layer Selection Section](#).



All other settings for the **Thin Elastic Layer, Interface** node are described in the documentation for [Thin Elastic Layer](#) in the Solid Mechanics interface.

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](#) 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.

LAYER SELECTION

Select the layer or interfaces based on this node is added under which type of spring feature. You can select individual layers or interfaces in this node which are allowed on the parent node.



All other settings for the **Predeformation** node are described in the documentation for [Predeformation](#) in the Solid Mechanics interface.

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](#).

LAYER SELECTION

Select the material layers to which the added mass contribution is to be applied.

[

TABLE 6-30: LAYER SELECTIONS; ADDED MASS

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Boundary	Default	Selectable	When a single layered material definition is selected
Edge	Default	Selectable	When a single layered material definition is selected.
Point	Default	Selectable	When a single layered material definition is selected.

For a general description of this section, see [The Layer Selection Section](#).



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

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.

LAYER SELECTION

Select the interfaces between the layers to which the added mass contribution is to be applied.

[

TABLE 6-31: LAYER SELECTIONS; ADDED MASS, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	All interfaces when a single layered material definition is selected
Edge	No	Selectable	All interfaces when a single layered material definition is selected.
Point	No	Selectable	All interfaces when a single layered material definition is selected.



All other settings for the **Added Mass, Interface** node are described in the documentation for [Added Mass](#) in the Solid Mechanics interface.

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.

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.

LAYER SELECTION

TABLE 6-32: LAYER SELECTIONS; RIGID CONNECTOR

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available

For a general description of this section, see [The Layer Selection Section](#)



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 edges will move as a single rigid object, irrespective of whether they are geometrically adjacent or not.

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.

LAYER SELECTION

[

TABLE 6-33: LAYER SELECTIONS; RIGID CONNECTOR, INTERFACE

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL INTERFACES
Boundary	No	Selectable	Top and bottom faces can be selected

For a general description of this section, see [The Layer Selection Section](#)



All other settings for the **Rigid Connector, Interface** node are described in the documentation for [Rigid Connector](#) in the Solid Mechanics interface.

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

LAYER SELECTION

Select the layer or interfaces based on this node is added under which type of rigid connector feature. You can select individual layers or interfaces in this node which are allowed on 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.

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.

LAYER SELECTION

[

TABLE 6-34: LAYER SELECTIONS; ATTACHMENT

SELECTION	ALL LAYERED MATERIALS	LAYERED MATERIAL LINK	SELECTION OF INDIVIDUAL LAYERS
Edge	Default	Selectable	Not available

For a general description of this section, see [The Layer Selection Section](#)



The **Attachment** node is available with the addition of 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 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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

LAYER SELECTION

Select the layer or interfaces based on this node is added under which type of attachment feature. You can select individual layers or interfaces in this node which are allowed on the parent node.

For a general description of this section, see [The Layer Selection Section](#).



All other settings for the **Thermal Expansion** node are described in the documentation for [Thermal Expansion \(Attachment\)](#) in the Solid Mechanics interface.

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 don't 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, **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.

For a general description of this section, see [The Layer Selection Section](#).

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)

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.

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 r - z plane.

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 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

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.

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 λ_n , it must be augmented by the normal deformation gradient \mathbf{F}_N to define the full deformation gradient tensor.

$$\mathbf{F}_N = \lambda_n \mathbf{n} \cdot \mathbf{N}^t$$

\mathbf{n} is the normal vector to the deformed membrane. The full deformation gradient tensor \mathbf{F} is the sum of tangential and normal deformation gradient tensors.

$$\mathbf{F} = \mathbf{F}_T + \mathbf{F}_N = \mathbf{I} - \mathbf{N} \cdot \mathbf{N}^t + \nabla_T \mathbf{u} + (1 + u_{nn}) \mathbf{n} \cdot \mathbf{N}^t$$

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 tangent plane 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:

$$N = d \cdot \bar{\mathbf{s}}$$

where $\bar{\mathbf{s}}$ 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 <http://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`.

THICKNESS

Define the **Thickness** d by entering a value or expression in the field. The default is 0.0001 m. Use the [Change Thickness](#) node to define a different thickness in parts of the membrane. The thickness can be variable if an expression is used.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasi-static**. Use **Quasi-static** to treat the elastic 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.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation** \mathbf{x}_{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.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field** \mathbf{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.

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**.



- [Boundary, Edge, Point, and Pair Nodes for the Membrane Interface](#)
- [Theory for the Membrane Interface](#)
- [Selecting Discretization](#)



Vibrating Membrane: Application Library path
Structural_Mechanics_Module/Verification_Examples/vibrating_membrane

Boundary, Edge, Point, and Pair Nodes for the Membrane Interface

The **Membrane Interface** has these boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users)..



In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Membrane interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

- The submenus at the **Boundary** level are **Material Models**, **Face and Volume Loads**, **Mass**, **Spring**, and **Damper**, and **Face Constraints**.
- The submenus at the **Edge** (3D) or **Points** (2D axisymmetric) level are **Mass**, **Spring**, and **Damper**, **More Constraints**, and **Pairs**.
- For 3D components, there is also a **Points** submenu.

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- [Initial Stress and Strain](#)
- [External Stress](#)
- [Linear Elastic Material](#)

These nodes are described for the Solid Mechanics interface:

- Added Mass
- Body Load¹
- Creep
- Damping
- Edge Load³
- External Stress
- Face Load³
- Fixed Constraint
- Free
- Gravity¹
- Hygroscopic Swelling
- Hyperelastic Material
- Initial Values
- Nonlinear Elastic Material
- Phase
- Plasticity
- Point Load
- Point Load (on Axis)
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement²
- Prescribed Velocity
- Ring Load
- Rotating Frame¹
- Spring Foundation
- Symmetry Plane
- Thermal Expansion (for Constraints)
- Thermal Expansion (for Materials)
- Viscoelasticity
- Viscoplasticity

¹ This is selected from the **Face and Volume Loads** submenu for this interface.

² At the boundary level, this is selected from the **Face Constraints** submenu for this interface.

³ Described for the Shell interface.



Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis
in the *COMSOL Multiphysics Reference Manual*



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic membrane and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [Damping](#)
- [Viscoelasticity](#)
- [Plasticity](#)
- [Creep](#)
- [Viscoplasticity](#)

Note: Some options are only available with certain COMSOL products (see <http://www.comsol.com/products/specifications/>)

COORDINATE SYSTEM SELECTION

The built in **Boundary System I** 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 , ν_{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** ρ uses values **From material**. For **User defined** enter another value or expression.



The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties ([Computing Mass Properties](#)).

Specification of Elastic Properties for Isotropic Materials

For an **Isotropic 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** ν .

- **Shear modulus** G .
- **Bulk modulus** K .
- **Lamé parameter** λ and **Lamé parameter** μ .
- **Pressure-wave speed** (longitudinal wave speed) c_p .
- **Shear-wave speed** (transverse wave speed) c_s . This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Specification of Elastic Properties for Orthotropic Materials

When **Orthotropic** is selected from the **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** ν , 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

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$



ν_{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**, \mathbf{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** button () and select **Advanced Physics Options**.

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

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](#).

COORDINATE SYSTEM SELECTION

The given initial stresses and strains are interpreted in this system.



For details about how local coordinate systems are used in the Membrane interface, see [Local Coordinate Systems](#).

INITIAL STRESS AND STRAIN

Specify the initial stress as the **Initial local in-plane force** N_0 and the initial strain as the **Initial local in-plane strain** ε_0 (dimensionless). 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>Nonlinear Elastic Material>Initial Stress and Strain

Ribbon

Physics tab with **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 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.



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 (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** drop-down list will contain all stress tensors announced by any physics interface, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** S_{ext} as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored—the orientation is handled internally. Choose a **Contribution type**—**Add to stress tensor** 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** σ_{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_{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

Beam

This chapter describes the Beam interface, which you find under the **Structural Mechanics** branch () when adding a physics interface.

In this chapter:

- [Theory for the Beam Interface](#)
- [The Beam Interface](#)

Theory for the Beam Interface

The [Beam Interface](#) theory is described in this section:

- [About Beams](#)
- [In-Plane Beams](#)
- [3D Beam](#)
- [Geometric Variables](#)
- [Shape Functions](#)
- [Geometric Nonlinearity](#)
- [Strain-Displacement/Rotation Relation](#)
- [Stress-Strain Relation](#)
- [Thermal Strain](#)
- [Hygroscopic Swelling](#)
- [Initial Load and Strain](#)
- [Implementation](#)
- [Stress Evaluation](#)
- [Common Cross Sections](#)



Beam Cross Section

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 lattice works, both planar and three-dimensional.

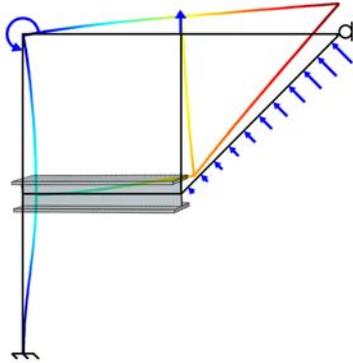
Beams can sustain forces and moments in any direction, both distributed and on individual nodes. The beam's ends and interconnections can be free, simply supported, or clamped. In fact, the simplified boundary conditions are usually responsible for most of the difference that can be found between a beam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

The Beam interface is based on the principle of virtual work. The resulting equation can equivalently be viewed as a weak formulation of an underlying PDE. The Beam

interface uses special shape function classes to define stresses and strains in the beams using either Euler-Bernoulli or Timoshenko theory.

In-Plane Beams

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



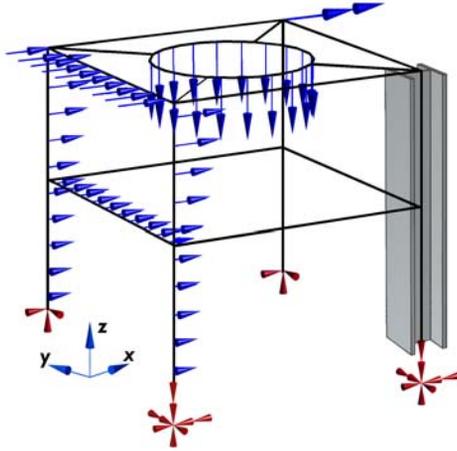
In-plane beams are defined on edges in 2D. They can be used separately or as stiffeners to 2D solid elements.

VARIABLES AND SPACE DIMENSIONS

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

3D Beam

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



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

VARIABLES AND SPACE DIMENSIONS

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

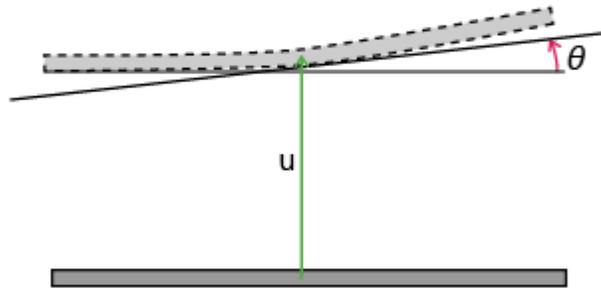


Figure 8-1: The degrees of freedom in the Beam interface,

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
μ_y	beam.muy	Max shear stress factor in local y-direction
κ_y	beam.kappay	Shear correction factor along local y-axis
r_{gy}	beam.rgy	Radius of gyration, local y-direction

TABLE 8-1: GEOMETRIC VARIABLES IN BEAM INTERFACE

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

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 is represented by cubic shape functions, usually called Hermitian shape functions. These will supply exact solutions to the underlying beam equations as long as distributed loads do not vary with position.

The shape functions for bending depend on whether Timoshenko theory is employed or not.

In the beam local system, the displacements, \mathbf{u} , and rotations, θ , are interpolated as

$$\begin{Bmatrix} \mathbf{u} \\ \theta \end{Bmatrix} = [\mathbf{N}] \begin{Bmatrix} \mathbf{u}_1 \\ \theta_1 \\ \mathbf{u}_2 \\ \theta_2 \end{Bmatrix}$$

where the subscript refers to the two nodes of the element, and \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 ξ , 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, θ . The rotation at the midpoint is approximated as the arithmetic mean of the rotations at the nodes,

$$\theta_M = \frac{\theta_1 + \theta_2}{2}$$

The rigid rotation is then represented by a rotation matrix \mathbf{R}_r , corresponding to this midpoint rotation. It is given by

$$\mathbf{R}_r = \mathbf{I} + \frac{\sin|\theta_M|}{|\theta_M|}\Theta_M + \frac{1 - \cos|\theta_M|}{|\theta_M|^2}\Theta_M^2$$

where Θ_M is the skew symmetric representation of the midpoint rotation vector.

$$\Theta_M = \begin{bmatrix} 0 & -\theta_{Mz} & \theta_{My} \\ \theta_{Mz} & 0 & -\theta_{Mx} \\ -\theta_{My} & \theta_{Mx} & 0 \end{bmatrix}$$

The axis directions of the co-rotated coordinate system can now be computed as

$$[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3] = \mathbf{R}_r [\mathbf{r}_1^0 \ \mathbf{r}_2^0 \ \mathbf{r}_3^0]$$

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 ξ ranges from 0 to 1, and \mathbf{X}_i denotes original node coordinates. \mathbf{x}_M is the midpoint position, computed as the average of the two nodes,

$$\mathbf{x}_M = \frac{1}{2}(\mathbf{X}_1 + \mathbf{u}_1 + \mathbf{X}_2 + \mathbf{u}_2)$$

In addition to the rigid body motion described so far, there are the local deformational displacements with respect to the local rotated beam axes. The deformational displacement can be computed as the difference between the current position and the rigid body position

$$\bar{\mathbf{u}} = \mathbf{X} + \mathbf{u} - \mathbf{x}_R = \mathbf{u} - \mathbf{u}_R$$

Here, and in the following, an over-bar 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

$$\varepsilon = z_1 \frac{\partial \theta_{1y}}{\partial s} - y_1 \frac{\partial \theta_{1z}}{\partial s} + \frac{\partial u_{axi}}{\partial s}$$

The coordinates from the beam center line 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 ε consists of thermal (ε_{th}), hygroscopic (ε_{hs}), initial (ε_i), and elastic strains (ε_{el})

$$\varepsilon = \varepsilon_{el} + \varepsilon_{th} + \varepsilon_{hs} + \varepsilon_i$$

Stress-Strain Relation

The stress-strain relation for the axial deformation in the beam is described by

$$\sigma = E \varepsilon_{el} + \sigma_i$$

where E is Young's modulus, and σ_i is the initial stress.

The stress strain relation for the torsional and shear deformation is

$$\tau = G \gamma$$

where τ is the shear stress, γ is the shear strain and G is the shear modulus. Often the material data is given in terms of Young's modulus and Poisson's ratio ν , in which case G is defined as

$$G = \frac{E}{2(1 + \nu)}$$

Thermal Strain

The temperature is assumed to vary linearly across the beam's cross section. For the 3D beam it becomes

$$T = T_m + T_{gz}z_1 + T_{gy}y_1$$

T_m is the temperature at the beam center line while T_{gz} and T_{gy} are the temperature gradients in the two local transversal directions. The thermal strain is thus

$$\epsilon_{th} = \alpha(T_m + T_{gz}z_1 + T_{gy}y_1 - T_{ref})$$

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

$$c_{mo} = c_m + c_{gz}z_1 + c_{gy}y_1$$

c_m is the concentration at the beam center line while c_{gz} and c_{gy} are the concentration gradients in the two local transversal directions. The strain from hygroscopic swelling is then

$$\epsilon_{th} = \beta_h(c_m + c_{gz}z_1 + c_{gy}y_1 - c_{mo,ref})$$

where β_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.

$$\begin{aligned}M_{yi} &= \int_A \sigma_i z_1 dA \\M_{zi} &= -\int_A \sigma_i y_1 dA \\N_i &= \int_A \sigma_i dA \\M_{xi} &= \int (\tau_{ixz} y_1 - \tau_{iyz} z_1) dA\end{aligned}$$

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.

$$\begin{aligned}T_{yi} &= \int_A \tau_{ixy} dA \\T_{zi} &= \int_A \tau_{ixz} dA\end{aligned}$$

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

$$\varepsilon_i = z_1 \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - y_1 \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i + \left(\frac{\partial u_{axi}}{\partial s} \right)_i$$

In 2D the z_1 dependent term disappears. As initial strain for the torsional degree of freedom, the derivative of the twist angle,

$$\left(\frac{\partial \theta_{1x}}{\partial s} \right)_i$$

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:

$$\delta W = \delta \int_V (-\varepsilon_{el} \sigma + \mathbf{u}^T \mathbf{F} dV) = 0$$

The beam elements are formulated in terms of the stress resultants (normal force, bending moments and twisting moment).

The normal force is defined as

$$\begin{aligned} N &= \int_A \sigma dA = \int_A (E\varepsilon_{el} + \sigma_i) dA = \\ & \int_A \left(E \left[\left(z_1 \frac{\partial \theta_{1y}}{\partial s} - y_1 \frac{\partial \theta_{1z}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right) - \left(z_1 \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - y_1 \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i + \left(\frac{\partial u_{axi}}{\partial s} \right)_i \right) - \right. \right. \\ & \left. \left. \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{m0,ref}) \right] + \sigma_i \right) dA = \\ & \int_A \left(E \left[\left(\frac{\partial u_{axi}}{\partial s} \right) - \left(\frac{\partial u_{axi}}{\partial s} \right)_i \right] - \alpha (T_m - T_{ref}) - \beta_h (c_m - c_{m0,ref}) \right) dA + \int_A \sigma_i dA = \\ & EA \left[\left(\frac{\partial u_{axi}}{\partial s} - \left(\frac{\partial u_{axi}}{\partial s} \right)_i \right) - \alpha (T_m - T_{ref}) - \beta_h (c_m - c_{m0,ref}) \right] + N_i \end{aligned}$$

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

$$\begin{aligned} M_{1y} &= \int_A \sigma z_1 dA = \int_A z_1 (E\varepsilon_{el} + \sigma_i) dA = \\ & \int_A z_1 \left(E \left[\left(z_1 \frac{\partial \theta_{1y}}{\partial s} - y_1 \frac{\partial \theta_{1z}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right) - \left(z_1 \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - y_1 \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i + \left(\frac{\partial u_{axi}}{\partial s} \right)_i \right) - \right. \right. \\ & \left. \left. \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{m0,ref}) \right] + \sigma_i \right) dA = \\ & \int_A z_1^2 \left(E \left[\frac{\partial \theta_{1y}}{\partial s} - \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - \alpha T_{gz} - \beta_h c_{gz} \right] \right) dA + \int_A \sigma_i z_1 dA = \\ & EI_{yy} \left[\frac{\partial \theta_{1y}}{\partial s} - \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - \alpha T_{gz} - \beta_h c_{gz} \right] + M_{1y} \end{aligned}$$

$$\begin{aligned}
M_{1z} &= -\int \sigma_y y_1 dA = -\int y_1 (E \varepsilon_{e1} + \sigma_i) dA = \\
&-\int y_1 \left(E \left[z_1 \frac{\partial \theta_{1y}}{\partial s} - y_1 \frac{\partial \theta_{1z}}{\partial s} + \frac{\partial u_{axi}}{\partial s} \right] - \left(z_1 \left(\frac{\partial \theta_{1y}}{\partial s} \right)_i - y_1 \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i + \left(\frac{\partial u_{axi}}{\partial s} \right)_i \right) \right. \\
&\left. - \alpha (T_m + T_{gz} z_1 + T_{gy} y_1 - T_{ref}) - \beta_h (c_m + c_{gz} z_1 + c_{gy} y_1 - c_{mo,ref}) \right] + \sigma_i) dA = \\
&-\int y_1^2 \left(E \left[-\frac{\partial \theta_{1z}}{\partial s} + \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i - \alpha T_{gy} - \beta_h c_{gy} \right] \right) dA - \int \sigma_i y_1 dA = \\
&EI_{zz} \left[\frac{\partial \theta_{1z}}{\partial s} - \left(\frac{\partial \theta_{1z}}{\partial s} \right)_i + \alpha T_{gy} + \beta_h c_{gy} \right] + M_{iz}
\end{aligned}$$

M_{1y} is present only in 3D, and so is the torsional moment M_{1x} 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

$$M_{1x} = GJ \left(\frac{\partial \theta_{1x}}{\partial s} - \left(\frac{\partial \theta_{1x}}{\partial s} \right)_i \right) + M_{ix}$$

Using the beam moment and normal force the expression for the virtual work becomes very compact:

$$\delta W = -\int_L \left(M_{1y} \delta \left(\frac{\partial \theta_{1y}}{\partial s} \right) + M_{1z} \delta \left(\frac{\partial \theta_{1z}}{\partial s} \right) + N \delta \left(\frac{\partial u_{axi}}{\partial s} \right) + M_{1x} \delta \left(\frac{\partial \theta_{1x}}{\partial s} \right) \right) dx$$

For 2D, the first and fourth terms are omitted. For the case of Timoshenko beam, there is also a shear stress contribution added,

$$\delta W = -\int_L (T_{1y} \delta \gamma_{xy} + T_{1z} \delta \gamma_{xz}) dx$$

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_{1x}, f_{1y}, f_{1z}) , which is defined as acting along the centerline, is then augmented by a twisting moment given by

$$m_{1x} = f_{1y}e_z - f_{1z}e_y$$

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

$$\sigma_{bk} = \frac{M_{1y}z_{1k}}{I_{yy}} - \frac{M_{1z}y_{1k}}{I_{zz}}$$

In 2D, only two points, specified by their local y -coordinates are used.

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

$$\sigma_{b, \max} = \frac{d_o \sqrt{M_{1y}^2 + M_{1z}^2}}{2I_{zz}}$$

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

$$\tau_{t, \max} = \frac{|M_{1x}|}{W_t}$$

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

$$\tau_{sz, \text{ave}} = \frac{T_{1z}}{A}$$

$$\tau_{sy, \text{ave}} = \frac{T_{1y}}{A}$$
(8-1)

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\tau_{sz, \max} = \mu_z \tau_{sz, \text{ave}}$$

$$\tau_{sy, \max} = \mu_y \tau_{sy, \text{ave}}$$
(8-2)

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

$$\tau_{xz, \max} = |\tau_{sz, \max}| + \tau_{t, \max}$$

$$\tau_{xy, \max} = |\tau_{sy, \max}| + \tau_{t, \max}$$

The maximum von Mises effective stress for the cross section is the defined as

$$\sigma_{\text{mises}} = \sqrt{\sigma_{\text{max}}^2 + 3\tau_{xy, \text{max}}^2 + 3\tau_{xz, \text{max}}^2}$$

Since the maximum values for the different stress components in general occur at different positions over the cross section, the effective stress thus computed is a conservative approximation.

Common Cross Sections

The cross section data for the common cross sections can be computed internally in COMSOL Multiphysics. In this section, the expressions used are summarized.



Beam Cross Section

RECTANGULAR SECTION

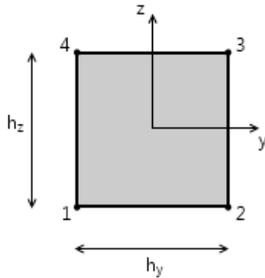


Figure 8-2: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

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	

TABLE 8-2: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
μ_y	1.5	
κ_y	5/6	
I_{yy}	$\frac{h_z^3 h_y}{12}$	
e_y	0	
μ_z	1.5	
κ_z	5/6	
J	$\frac{h_y^2 h_z^2 q}{3} \left(1 - \sum_{n=1}^z \frac{192q}{\pi^5 (2n-1)^5} \tanh\left(\frac{(2n-1)\pi}{2q}\right) \right)$	$q = \min(h_y/h_z, h_z/h_y)$
W_t	$\frac{J}{\sqrt{h_y h_z q} \left(1 - \sum_{n=1}^2 \frac{8}{\pi^2 (2n-1)^2 \cosh\left(\frac{(2n-1)\pi}{2q}\right)} \right)}$	$q = \min(h_y/h_z, h_z/h_y)$
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

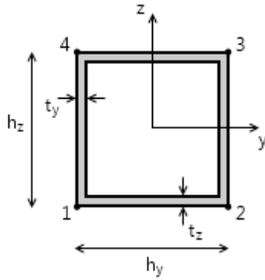


Figure 8-3: Geometry of a box shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

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	
μ_y	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - 2t_z))A}{16t_z I_{zz}}$	
κ_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	
μ_z	$\frac{(h_z^2 h_y - (h_z - 2t_z)^2 (h_y - 2t_y))A}{16t_y I_{yy}}$	
κ_z	$\frac{2h_z t_y}{A}$	

TABLE 8-3: BOX SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
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	$\left(\frac{h_y}{2}, \frac{h_z}{2}\right)$	
p_4	$\left(\frac{-h_y}{2}, \frac{h_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

CIRCULAR SECTION

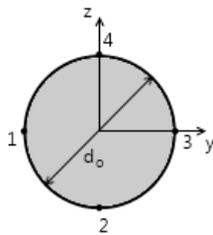


Figure 8-4: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

C

TABLE 8-4: CIRCULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
A	$\frac{\pi d_o^2}{4}$	
I_{zz}	$\frac{\pi d_o^4}{64}$	
e_z	0	
μ_y	$\frac{4}{3}$	
κ_y	0.9	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	
κ_z	κ_y	
J	$\frac{\pi d_o^4}{32}$	
W_t	$\frac{\pi d_o^3}{16}$	
p_1	$\left(-\frac{d_o}{2}, 0\right)$	
p_2	$\left(0, -\frac{d_o}{2}\right)$	
p_3	$\left(\frac{d_o}{2}, 0\right)$	
p_4	$\left(0, \frac{d_o}{2}\right)$	
r_e	$\frac{d_o}{2}$	

PIPE SECTION

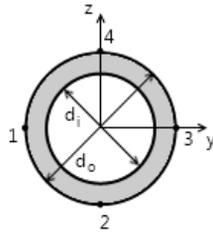


Figure 8-5: Geometry of a pipe cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

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	
μ_y	$\frac{(d_o^3 - d_i^3)A}{12(d_o - d_i)I_{zz}}$	
κ_y	$1 + \frac{d_i}{d_o}$	
I_{yy}	I_{zz}	
e_y	0	
μ_z	μ_y	
κ_z	κ_y	

TABLE 8-5: PIPE SECTION CONSTANTS

PROPERTY	FORMULA	REMARKS
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	$\left(\frac{d_o}{2}, 0\right)$	
p_4	$\left(0, \frac{d_o}{2}\right)$	
r_e	$\frac{d_o}{2}$	

H-PROFILE SECTION

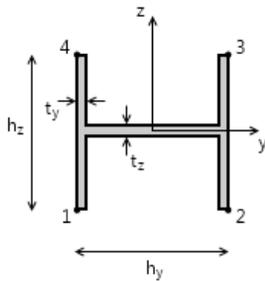


Figure 8-6: Geometry of an H-profile cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

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	
μ_y	$\frac{[4h_z t_y (h_y - t_y) + t_z (h_y - 2t_y)^2]A}{8t_z I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{2t_y h_z^3 + t_z^3 (h_y - 2t_y)}{12}$	
e_y	0	
μ_z	$\frac{(h_z^2 - t_z^2)A}{8I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{2t_y^3 h_z + t_z^3 (h_y - 2t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, \frac{-h_z}{2}\right)$	
p_2	$\left(\frac{h_y}{2}, \frac{-h_z}{2}\right)$	
p_3	$\left(\frac{h_y}{2}, \frac{h_z}{2}\right)$	

TABLE 8-6: H-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
p_4	$\left(\frac{-h_y}{2}, \frac{h_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

U-PROFILE SECTION

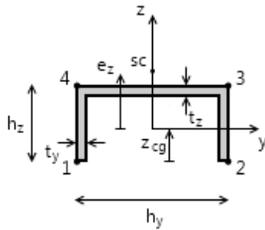


Figure 8-7: Geometry of a U-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 8-7: U-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
A	$h_y t_z + 2(h_z - t_z) t_y$	
I_{zz}	$\frac{t_z h_y^3 + 2t_y^3 (h_z - t_z)}{12} + \frac{t_y (h_z - t_z) (h_y - t_y)^2}{2}$	
z_{CG}	$\frac{h_y t_z (2h_z - t_z) + 2t_y (h_z - t_z)^2}{2A}$	
e_z	$h_z - \frac{t_z}{2} + \frac{t_y (2h_z - t_z)^2 (h_y - t_y)^2}{16I_{zz}} - z_{CG}$	
μ_y	$\frac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - t_z)) A}{8t_z I_{zz}}$	

TABLE 8-7: U-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{8t_y h_z^3 + t_z^3 (h_y - 2t_y) + 3t_z (h_y - 2t_y)(2h_z - t_z)^2}{12} - z_{CG}^2 A$	
e_y	0	
μ_z	$\frac{z_{CG}^2 A}{2I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{2t_y^3 h_z + t_z^3 (h_y - 2t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, -z_{CG}\right)$	
p_2	$\left(\frac{h_y}{2}, -z_{CG}\right)$	
p_3	$\left(\frac{h_y}{2}, h_z - z_{CG}\right)$	
p_4	$\left(\frac{-h_y}{2}, h_z - z_{CG}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

T-PROFILE SECTION

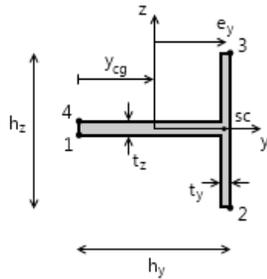


Figure 8-8: Geometry of a T-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

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	
μ_y	$\frac{y_{CG}^2 A}{2I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{t_y h_z^3 + t_z^3 (h_y - t_y)}{12}$	
e_y	$h_y - \frac{t_y}{2} - y_{CG}$	

TABLE 8-8: T-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
μ_z	$\frac{(h_z^2 - t_z^2)A}{8I_{yy}}$	
κ_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

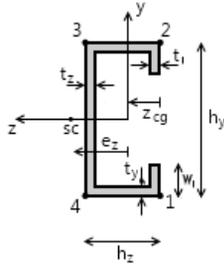


Figure 8-9: Geometry of a C-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

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
μ_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
κ_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	
μ_z	$\frac{(t_y z_{CG}^2 + (w_1 - t_y)t_1(2z_{CG} - t_1))A}{2t_y I_{yy}}$	
κ_z	$\frac{5h_z t_y}{3A}$	
J	$\frac{t_z^3 h_y + 2t_y^3 (h_z - t_z) + 2t_1^3 (w_1 - t_y)}{3}$	Thin-walled approximation
W_t	$\frac{J}{\max(t_y, t_z, t_1)}$	Thin-walled approximation
p_1	$\left(\frac{-h_y}{2}, -z_{CG} \right)$	
p_2	$\left(\frac{h_y}{2}, -z_{CG} \right)$	
p_3	$\left(\frac{h_y}{2}, h_z - z_{CG} \right)$	
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

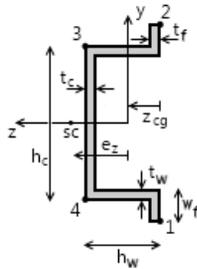


Figure 8-10: Geometry of a Hat cross section. Also displays in COMSOL Multiphysics when this option is selected.

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
μ_y	$[h_c^2 t_c + 4(h_w - t_c)t_w(h_c - t_w) + 4(w_f - t_w)t_f(h_c + w_f)] \frac{A}{8t_c I_{zz}}$	
κ_y	$\frac{(h_c - 2t_w)t_c}{A}$	
I_{yy}	$\frac{t_c^3 h_c + 8(h_w - t_c)^3 t_w + 8t_f^3 (w_f - t_w)}{12} + \frac{3t_c h_c (2h_w - t_c)^2}{12} - z_{CG}^2 A$	
e_y	0	
μ_z	$\frac{(t_w z_{CG}^2 + (w_f - t_w)t_f(2z_{CG} - t_f))A}{2t_w I_{yy}}$	
κ_z	$\frac{5h_w t_w}{3A}$	
J	$\frac{t_c^3 h_c + 2t_w^3 (h_w - t_c) + 2t_f^3 (w_f - t_w)}{3}$	Thin-walled approximation
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 an 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 [Theory for 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`.

BEAM FORMULATION

Select **Euler-Bernoulli** or **Timoshenko** to use the appropriate beam theory.



For COMSOL Multiphysics models created in versions 4.3b and earlier, the only option available is **Euler-Bernoulli**. See the [Backward Compatibility](#) section to make changes to such a model.

BACKWARD COMPATIBILITY



The formulation of the beam element was significantly changed as of COMSOL Multiphysics version 4.4. For this reason, by default, models created in previous versions default to using the **Euler-Bernoulli** theory.

This section only displays if a model was created in an earlier version. You also need to click the **Show** button () and select **Advanced Physics Options** to edit the section and change the Beam Formulation to **Timoshenko** (if required).

Click to clear the **Use pre 4.4 formulation** check box to convert a model into the new formulation. For most models, the only difference is that it will then be possible to select **Timoshenko** theory in the **Beam Formulation** section.

There are, however, some situations where you need to make additional changes to your model when migrating to the new formulation:

- If you have made manual adjustments to the solver sequence, such as scaling the dependent variables or adjusting groups in a segregated solver. These settings might have to be redone.
- If you are using multiple physics interfaces and use the same degree of freedom names for the displacements in the Beam interface and other structural mechanics interfaces. In this case, you need to rename the degrees of freedom in the Beam interface and create a different type of connection between the beams and the other physics interface, for example using a [Solid Connection](#) or [Shell Connection](#).

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the default coordinates for the **Reference point for moment computation** \mathbf{x}_{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.

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 ($u, v, \text{ and } w$) in 3D.
- The rotation angle θ , which has one component in 2D (th) and three components in 3D ($thx, thy, \text{ and } thz$).

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 (`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.

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”).



- [Boundary, Edge, Point, and Pair Nodes for the Beam Interface](#)
- [Theory for the Beam Interface](#)
- [Theory for the Beam Cross Section 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**
- **Connections**
- **Line Constraints.**

The submenus at the **Point** level are

- **Mass, Spring, and Damper**
- **Connections**
- **More Constraints**
- **Pairs.**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- [Antisymmetry](#)
- [Attachment](#)
- [Cross Section Data](#)
- [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 Acceleration](#)
- [Prescribed Displacement/Rotation](#)
- [Prescribed Velocity](#)
- [Rigid Connector](#)
- [Section Orientation](#)
- [Shell Connection](#)
- [Solid Connection](#)
- [Symmetry](#)
- [Thermal Expansion \(for Materials\)](#)

These nodes are described for the Solid Mechanics interface:

- [Added Mass](#)
- [Damping](#)
- [Fixed Constraint](#)
- [Free](#)
- [Gravity¹](#)
- [Predeformation](#)
- [Rigid Domain](#)
- [Rotating Frame¹](#)
- [Spring Foundation](#)

¹ This is selected from the **Line and Volume Loads** submenu for this interface.



If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence. The exception is that the “Pinned” and “No Rotation” boundary conditions do not override each other since the degrees of freedom that they constrain are mutually exclusive.



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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** θ
- **Angular velocity** $\frac{d\theta}{dt}$

Context Menus

Beam>Initial Values

Beam>Edges>Initial Values

Ribbon

Physics tab with **Solid Mechanics** selected:

Edges>Beam>Initial Values

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/Civil_Engineering/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	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	κ_y	1
3D only	Distance to shear center in local z-direction	e_z	m
3D only	Moment of inertia about local y-axis	I_{yy}	m^4
3D only	Distance to shear center in local y-direction	e_y	m
3D only	Torsional constant	J	m^4
3D only, Timoshenko beam	Shear correction factor along local z-axis	κ_z	1



For 3D models, the orientation of the cross section is given in [Section Orientation](#). If the beam's cross section is a square or circle (solid or tube), the area moments of inertia are the same independent of direction, so the beam is totally symmetric and the orientation of the principal axes of the cross section is not a problem unless you are interested in looking at results defined using the local coordinate system. Such results are bending moments, shear forces, local displacements and rotations.

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 8-1](#) and [Equation 8-2](#).

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	μ_y	l
3D only	Section height in local z direction	h_z	m
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	l

From Specified Points

For **From specified points** enter values in the **Evaluation points in local system** table as needed for the space dimension. Then enter the following parameters in the applicable fields.

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Max shear stress factor in local y direction	μ_y	l
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	l

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 8-11](#).

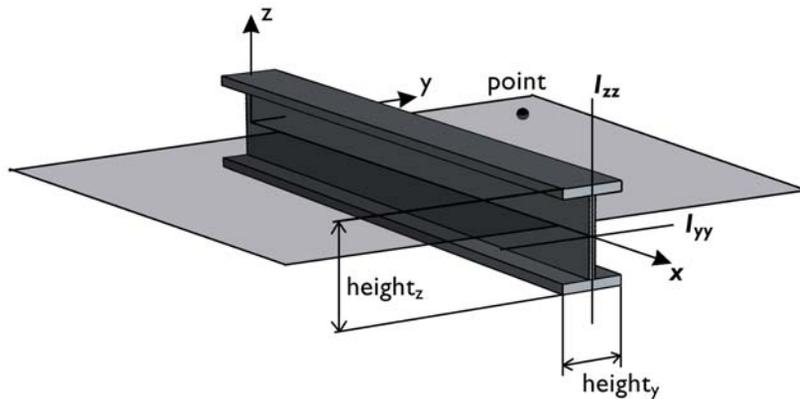


Figure 8-11: Local beam coordinate system defined by a reference point.

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 a **Orientation vector defining local y direction** V and optionally the **Rotation of vector around beam axis** ϕ . The beam orientation is defined similarly to what is described above, with the difference that in this case the direction vector is explicitly defined whereas when an orientation point is used, the direction vector is obtained as the vector from the beam axis to the specified point. The local coordinate system (\mathbf{e}_{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

Ribbon

Physics tab with **Cross Section Data** node selected in the model tree:

Attributes>Section Orientation

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic beam and an interface for defining the elastic material properties.

By adding the following subnodes to the **Linear Elastic Material** node you can incorporate many other effects:

- [Thermal Expansion \(for Materials\)](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [Damping](#)

LINEAR ELASTIC MATERIAL

Define the linear elastic material properties.

Specification of Elastic Properties for Isotropic Materials

From the **Specify** list, select a pair of elastic properties for an isotropic material. Select:

- **Young's modulus and Poisson's ratio** to specify Young's modulus (elastic modulus) E and Poisson's ratio ν . Poisson's ratio is used for computing the torsional stiffness, and is thus important only for 3D beams.
- **Young's modulus and shear modulus** to specify Young's modulus (elastic modulus) E and the shear modulus G . The shear modulus is used for computing the torsional stiffness, and is thus important only for 3D beams.
- **Bulk modulus and shear modulus** to specify the bulk modulus K and the shear modulus G .

- **Lamé parameters** to specify the Lamé parameters λ and μ .
- **Pressure-wave and shear-wave speeds** to specify the pressure-wave speed (longitudinal wave speed) c_p and the shear-wave speed (transverse wave speed) c_s .



This is the wave speed for a solid continuum. In a truss or beam element, the actual speed with which a longitudinal wave travels is lower than the value given. When using this type of input the density must also be given.

For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Each of these pairs define the elastic properties, and it is possible to convert from one set of properties to another.

Density

Define the **Density** ρ of the material. Select **From material** to take the value from the material or **User defined** to enter a value for the density.



The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties ([Computing Mass Properties](#)).



Thermally Loaded Beam: Application Library path
Structural_Mechanics_Module/Verification_Examples/thermally_loaded_beam

LOCATION IN USER INTERFACE

Context Menus

Beam>Material Models>Linear Elastic Material

Ribbon

Physics tab with **Beam** selected:

Edges>Material Models>Linear Elastic Material

Thermal Expansion (for Materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature. The thermal strain depends on the coefficient of thermal expansion (CTE) α , the temperature T , and the strain-free reference temperature T_{ref} as

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

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_{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.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.

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 [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain.

From the **Coefficient of thermal expansion** α list, select **From material** to use the coefficient of thermal expansion from the material, or **User defined** to enter a value or expression for α .

THERMAL BENDING

Enter the **Temperature gradient in local y-direction** T_{gy} (in 2D and 3D) and in the **Temperature gradient in local z-direction** T_{gz} (in 3D), which affects the thermal bending. If beam cross section dimensions have been defined at **Bending stress evaluation points** — **From section heights**, these could be used in an expression containing the temperature difference.

LOCATION IN USER INTERFACE

Context Menus

Beam>Linear Elastic Material>Thermal Expansion

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes>Thermal Expansion

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\epsilon_{hs} = \beta_h(c_{mo} - c_{mo,ref})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the transverse directions of the beam. The concentration is then assumed to vary linearly through the thickness.

MODEL INPUTS

From the **Concentration** c list, select an existing concentration variable from another physics interface, if any concentration variables exist. For **User defined** enter a value or expression for the concentration. This is the centerline concentration of the beam, controlling the axial part of the hygroscopic swelling.

The unit for the input depends on the setting of **Concentration type** in the **Hygroscopic Swelling Properties** section. Only concentration variables having the chosen physical dimension are available in the **Concentration** list.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** (the default) or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref} . This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as **Concentration type**, also enter the **Molar mass** of the fluid, M_m . The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_h uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of hygroscopic swelling tensor β_h . The default value for the **User defined** case is $1.5e-4 \text{ m}^3/\text{kg}$.

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 the **Linear Elastic Material** 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** θ_{siz} and for 3D models, and θ_{siy} .
- *For 3D models:* **Initial twist** θ_{six} .
- *For Timoshenko beam:* **Initial shear strain** γ_{siy} and for 3D models, γ_{siz} .



For details about initial stresses and strains, see [Inelastic Strain Contributions](#) and [Initial Stresses and Strains](#).

Context Menus

Beam>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 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.



In many cases **External Stress** and [Initial Stress and Strain](#) are interchangeable when prescribing stresses. In **Initial Stress and Strain**, the given stress is however always added to the stress tensor.

EXTERNAL STRESS

Specify the external stress as section forces:

- **External axial force** N_{ext} .
- **External bending moment** $M_{z,\text{ext}}$ and for 3D models, $M_{y,\text{ext}}$.
- *For 3D models:* **External torsional moment** $M_{x,\text{ext}}$.
- *For Timoshenko beam:* **External shear force** $T_{y,\text{ext}}$ and for 3D models, $T_{z,\text{ext}}$.

For each quantity, you can select **User defined**, in which case you enter values or expressions explicitly. The drop-down 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.

Context Menus

Beam>Linear Elastic Material>External Stress

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes>External Stress

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.

- 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

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 u_0 , v_0 , and for 3D models, w_0 .

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 θ_{0z} .



For 3D models, select a prescribed rotation from the list — **Free** or **Rotation**. Select:

- **Free** (the default) to leave the rotations unconstrained.
- **Rotation** to activate a prescribed rotation. Enter values or expressions for the prescribed rotation vector θ . When the study is geometrically linear, you directly prescribe the individual components of the rotation vector. Under **For geometric linearity**, select one or several of the **Free rotation around x direction**, **Free rotation around y direction**, and **Free rotation around z direction** 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, θ should be interpreted as a rotation vector in the given coordinate system. The norm of the vector is the angle of rotation, and the orientation of the axis of rotation is given by the vector components.



In a geometrically nonlinear analysis in 3D, you should prescribe all three components of the rotation vector. Prescribing only one or two components may not give unique results, since finite rotations are not commutative.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.



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)

Ribbon

Physics tab with **Beam** selected:

Edges>Line Constraints>Prescribed Displacement

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** can not be used as a weak constraint.
-

Location in User Interface

Context Menus

Beam>Line Constraints>Prescribed Velocity

Beam>More Constraints>Prescribed Velocity (Point)

Ribbon

Physics tab with **Beam** 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. With this boundary condition it is possible to prescribe a 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** can not 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

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** button () and select **Advanced Physics Options**.



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)

Ribbon

Physics tab with **Beam** selected:

Edges>Line Constraints>Pinned

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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from edge** to take the thermal expansion data from the edge being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the edge itself.

When **Inherit from edge** is not selected, enter:

- A value or expression for **Temperature** T , specifying the temperature distribution of the surrounding material. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** α . As a default, values **From material** are used. This requires that a material has been assigned to the points where the constraint is active.
For **User defined** enter the coefficient of thermal expansion α .
- A value or expression for the **Volume reference temperature** T_{ref} which is the temperature at which there are no thermal displacements at the constraints.

Enter the coordinates of the **Reference point**, the point where the displacement and rotation 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

Beam>Fixed>Thermal Expansion

Beam>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** button () and select **Advanced Physics Options**.



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)

Ribbon

Physics tab with **Beam** selected:

Edges>Line Constraints>No Rotation

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 drop-down 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** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)



[Symmetry Constraints](#)



You can activate and deactivate this boundary condition by assigning it to a constraint group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam>Line Constraints>Symmetry

Beam>More Constraints>Symmetry (Point)

Ribbon

Physics tab with **Beam** 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

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** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
 - [Weak Constraints](#)
 - [Constraint Settings](#)
-



Symmetry Constraints



You can activate and deactivate this boundary condition by assigning it to a constraint group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.

LOCATION IN USER INTERFACE

Context Menus

Beam>Line Constraints>Antisymmetry (Line)

Beam>More Constraints>Antisymmetry (Point)

Ribbon

Physics tab with **Beam** selected:

Edges>Line Constraints>Antisymmetry

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**—**Frce per unit length** (the default), **Force per unit volume**, or **Total force**. Enter values or expressions for the components.

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 ³	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 \mathbf{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

Beam Rotor>Edge Load

Ribbon

Physics tab with **Beam** selected:

Edges>Line and Volume Loads>Edge Load

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_{Iz} (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

Beam Rotor> Point Load

Ribbon

Physics tab with **Beam** 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

Ribbon

Physics tab with **Beam** 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** α_{dM} . This is the mass proportional term of a Rayleigh damping.

LOCATION IN USER INTERFACE

Context Menus

Beam>Point Mass>Point Mass Damping

Ribbon

Physics tab with **Point Mass** node selected in the model tree:

Attributes>Point Mass Damping



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.

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, 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.

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. The center of rotation is located at the centroid of the selected points, which do not need to be related to the points 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 **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.

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_{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** ϕ_0 . For 3D components also enter an **Axis of rotation** Ω for the **x**, **y**, and **z** coordinates.



- You can add a [Harmonic Perturbation](#) subnode for specifying a harmonic variation of the values of the prescribed displacements and rotations in a frequency domain analysis of perturbation type.
- You can activate and deactivate the rigid connector by assigning it to a constraint group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.
- You can assign the value of the prescribed displacement and rotation to a load group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.



- [Rigid Connector Theory](#)
- [Harmonic Perturbation](#)
- [Load Cases](#)

LOCATION IN USER INTERFACE

Context Menus

Beam>Connections>Rigid Connector

Ribbon

Physics tab with **Beam** selected:

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 the addition of 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

Beam>Connections>Attachment

Ribbon

Physics tab with **Beam** selected:

Points>Connections>Attachment

Phase

You can add a **Phase** subnode to nodes which define a load in order to prescribe the phase angle in a frequency domain analysis.

For modeling the frequency response the physics interface splits the harmonic load into two parameters:

- The amplitude, F , which is specified in the node for the load.
- The phase (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

PHASE

Add the phase load \mathbf{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 \mathbf{M}_{ph} for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

LOCATION IN USER INTERFACE

Context Menus

Beam>Edge Load>Phase

Beam>Point Load>Phase

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

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 Sections 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, respectively.



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)	Area
bcs.CGx	Center of gravity, x coordinate	Implicit, since it affects the positioning of the beam centerline.	Center of Gravity
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 (ϕ)	Directions of Principal Axes
bcs.mu1	Max shear stress factor, x1 direction	Max shear stress factor in local y/z direction (μ_y/μ_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
bcs.Cw	Warping constant	-	Warping
bcs.Kw	Warping section modulus	-	
bcs.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
bcs.sN	Stress from axial force	Axial Stress
bcs.sM1	Bending stress from moment around x1 axis	Bending Axial Stresses
bcs.sM2	Bending stress from moment around x2 axis	

TABLE 9-2: VARIABLES CONTAINING STRESS DISTRIBUTIONS

VARIABLE	DESCRIPTION AND	LINK TO THEORY SECTION
bcs.tT1x	Shear stress from force in x1 direction, x component	Bending Shear Stresses
bcs.tT1y	Shear stress from force in x1 direction, y component	
bcs.resT1	Shear stress from force in x1 direction, resultant	
bcs.tT2x	Shear stress from force in x2 direction, x component	
bcs.tT2y	Shear stress from force in x2 direction, y component	
bcs.resT2	Shear stress from force in x2 direction, resultant	
bcs.tMtx	Shear stress from twisting moment, x component	Torsional Shear Stresses
bcs.tMty	Shear stress from twisting moment, y component	
bcs.resMt	Shear stress from twisting moment, resultant	
bcs.mises	von Mises effective stress	Effective Stress
bcs.spN	Principal stresses; N=1,2,3	
bcs.INs	N:th invariant of stress tensor; N=1,2,3	
bcs.IINs	N:th invariant of stress tensor deviator; N=2,3	
bcs.thetaL	Lode angle for stress tensor	

COMPUTING WARPING DISPLACEMENT

You can also compute the axial warping displacement by entering the axial twist of the beam in the **Twist** section. The theory is described under [Warping](#). The axial displacement is stored in the variable `bcs.u`.

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 α . From the definition of Mohr's circle, the angle is:

$$\alpha = \frac{1}{2} \text{atan}\left(\frac{-2I_{xy}}{I_{xx} - I_{yy}}\right)$$

When implemented using the atan2 function, the angle can be correctly evaluated for all rotations, and returns in the interval $-\pi < \alpha < \pi$.

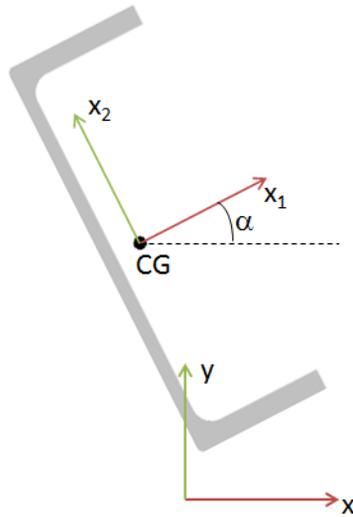


Figure 9-1: Local coordinate system and rotation angle.

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 κ_1 is the shear correction factor for a shear force in the x_1 direction. Thus, for a shear flexible beam, the constitutive relation for the average shear is

$$\bar{\gamma} = \frac{\bar{\tau}}{\kappa G} = \frac{T}{\kappa GA} = \frac{T}{GA_s}$$

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 gives 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 ψ , through:

$$\begin{aligned}\tau_{1z} &= \psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2} \\ \tau_{2z} &= \psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1}\end{aligned}$$

Insertion of this assumption into the third equilibrium equation gives the Poisson type equation:

$$\Delta\Psi = -\left(\frac{T_2x_2}{I_1} + \frac{T_1x_1}{I_2}\right) \quad (9-2)$$

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:

$$\Delta\tau_{1z} + \frac{1}{1+\nu} \frac{T_1}{I_2} = 0$$

$$\Delta\tau_{2z} + \frac{1}{1+\nu} \frac{T_2}{I_1} = 0$$

Inserting the assumed stress components gives:

$$\Delta\Psi_{,1} + \frac{\nu T_1}{(1+\nu)I_2} + \frac{1}{1+\nu} \frac{T_1}{I_2} = \Delta\Psi_{,1} + \frac{T_1}{I_2} = 0$$

$$\Delta\Psi_{,2} + \frac{\nu T_2}{(1+\nu)I_1} + \frac{1}{1+\nu} \frac{T_2}{I_1} = \Delta\Psi_{,2} + \frac{T_2}{I_1} = 0$$

Integration of the first equation with respect to x_1 and the second equation with respect to x_2 gives:

$$\Delta\Psi + \frac{T_1x_1}{I_2} + f_2(x_2) = 0$$

$$\Delta\Psi + \frac{T_2x_2}{I_1} + f_1(x_1) = 0$$

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 ψ , 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:

$$\begin{aligned}
 \tau_{1z}^{(1)} &= \psi_{,1}^{(1)} + \frac{\nu x_2^2}{2(1+\nu)I_2} \\
 \tau_{2z}^{(1)} &= \psi_{,2}^{(1)}
 \end{aligned}$$

The corresponding problem for the x_2 direction is:

$$\Delta \psi^{(2)} = -\frac{x_2}{I_1}$$

$$\nabla \psi^{(2)} \cdot \mathbf{n} = \frac{-vx_1^2}{2(1+\nu)I_1} n_1$$

$$\tau_{1z}^{(2)} = \psi_{,1}^{(2)}$$

$$\tau_{2z}^{(2)} = \psi_{,2}^{(2)} + \frac{vx_1^2}{2(1+\nu)I_1}$$

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 ϕ is the stress function. For a singly connected region the boundary condition is $\phi = 0$ along the whole boundary. Having solved this problem the torsional rigidity can be computed as:

$$J = 2 \int_A \phi dA$$

The shear stresses are defined as:

$$\tau_{xz} = \frac{\partial \phi}{\partial y}$$

$$\tau_{yz} = -\frac{\partial \phi}{\partial x}$$

The torsional modulus can be determined as:

$$W_t = \frac{J}{\max(|\nabla\phi|)}$$

In the case that there are internal holes in the section the situation is slightly more complex. The $\phi = 0$ condition is now applicable only to the 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 Γ_i . This is trivially fulfilled for the in-plane displacements, but the out-of-plane displacement, w , generates the necessary equations to determine H_i :

$$\begin{aligned} 0 &= \oint_{\Gamma_i} dw = \oint_{\Gamma_i} \left[\frac{\partial w}{\partial x} dx + \frac{\partial w}{\partial y} dy \right] = \oint_{\Gamma_i} \left[\left(\gamma_{xz} - \left(\frac{\partial u}{\partial z} \right) \right) dx + \left(\gamma_{yz} - \left(\frac{\partial v}{\partial z} \right) \right) dy \right] \\ &= \oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y \right) dx + \left(-\frac{\partial \phi}{\partial x} - x \right) dy \right] = \oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y \right) t_x - \left(\frac{\partial \phi}{\partial x} + x \right) t_y \right] ds \end{aligned}$$

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 ϕ depends linearly on the yet unknown variables H_i , the values of which can be solved by adding one equation:

$$\oint_{\Gamma_i} \left[\left(\frac{\partial \phi}{\partial y} + y \right) t_x - \left(\frac{\partial \phi}{\partial x} + x \right) t_y \right] ds = 0$$

for each hole.

The expression for the torsional rigidity must in this case be augmented to:

$$J = 2 \int_A \phi dA + 2 \sum_i H_i A_i$$

where A_i is the area of hole i .

WARPING

The warping properties of the cross section are not used by the Beam interface in COMSOL Multiphysics, since an assumption of pure 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, ϕ_{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](#)
- [Effective 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:

$$\tau_{Mt} = \sqrt{\tau_{Mtx}^2 + \tau_{Mty}^2}$$

EFFECTIVE STRESS

The von Mises effective 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** ν (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**, ϕ_{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** σ_{ts} .
- When **Failure Criterion** is **Tresca Isotropic**, enter **Tensile strength** σ_{ts} .
- When **Failure Criterion** is **Rankine Isotropic**, enter **Tensile strength** σ_{ts} and **Compressive strength** σ_{cs} .
- When **Failure Criterion** is **Bresler-Pister Isotropic**, enter **Tensile strength** σ_{ts} , **Compressive strength** σ_{cs} , and **Biaxial compressive strength** σ_{bc} .
- When **Failure Criterion** is **Willam-Warnke Isotropic**, enter **Tensile strength** σ_{ts} , **Compressive strength** σ_{cs} , and **Biaxial compressive strength** σ_{bc} .

- When **Failure Criterion** is **Ottosen Isotropic**, enter the **Compressive strength** σ_{cs} , **Ottosen parameters** a and b , the **Size factor** k_1 , and the **Shape factor** k_2 .
- When **Failure Criterion** is **User defined**, you enter two expressions describing the **Failure criterion** $g(S)$, used in the failure index, and the **Safety factor** $s_f(S)$ respectively. 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 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 $E0 * (1 - 0.9999 * (\text{truss.en} < 0))$.

- 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 ϵ_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 ϵ_{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

$$\epsilon_{ijT} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} \Big|_T + \frac{\partial u_j}{\partial x_i} \Big|_T + \frac{\partial u_k}{\partial x_i} \Big|_T \cdot \frac{\partial u_k}{\partial x_j} \Big|_T \right)$$

The engineering strain tensor used for small displacements is defined as

$$\epsilon_{ijT} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} \Big|_T + \frac{\partial u_j}{\partial x_i} \Big|_T \right) \quad (10-1)$$

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 lower case letters (x, y, z). If a Solid Mechanics or Membrane interface is present in the same model, then it becomes necessary to make a difference between the material frame and the spatial frame ([Material and Spatial Coordinates](#)). In this case the coordinates in the Truss interface change to (X, Y, Z).

STRESS-STRAIN RELATION

The constitutive relation for a truss is uniaxial. The axial stress, σ_n , is computed as

$$\sigma_n = \sigma_{ex} + E\varepsilon_{n,el} = \sigma_{ex} + E(\varepsilon_n - \varepsilon_{inel})$$

$$\varepsilon_{inel} = \varepsilon_0 + \varepsilon_{th} + \varepsilon_{hs} + \varepsilon_{pl}$$

$$\sigma_{ex} = \sigma_0 + \sigma_{ext}$$

where

- E is the modulus of elasticity
- ε_n is the total axial strain
- $\varepsilon_{n,el}$ is the elastic axial strain
- ε_{inel} is the sum of all inelastic strain contributions:
 - Initial strain, ε_0
 - Thermal strain, ε_{th}
 - Hygroscopic strain, ε_{hs}
 - Plastic strain, ε_{pl}
- σ_{ex} is the sum of all extra strain contributions:
 - Initial stress σ_0
 - External stress σ_{ext}

In a geometrically nonlinear analysis, this equation should be interpreted as a relation between Second Piola-Kirchhoff stresses and Green-Lagrange strains.

For output, the First Piola-Kirchhoff stress P_n is computed from the Second Piola-Kirchhoff stress using

$$P_n = S_n \cdot s'$$

where s' is the ratio between current and initial length. The axial force in the element is then computed as

$$N = P_n \cdot A_0$$

where A_0 is the undeformed cross-section area. The engineering (Cauchy) stress is defined by

$$\sigma_n = P_n \frac{A_0}{A}$$

where A is the deformed area of the element. For a geometrically linear analysis, the change in area is ignored, so that $A = A_0$.

For a geometrically nonlinear analysis, the area change is computed based on an assumption about a linear elastic material with Poisson's ratio ν . The area change is

$$\frac{A_0}{A} = (1 - 2\nu\varepsilon_n)^{-1}$$

This is the only occasion where the Truss interface makes use of the value of ν .

In a geometrically linear analysis all the stress representations have the same value.

IMPLEMENTATION

Using the principle of virtual work results in the following weak formulation

$$\delta W = d \int_V (-\varepsilon_n \sigma_n + \mathbf{u}^t \mathbf{F}_V) dV + \sum_i \mathbf{u}^t \mathbf{F}_{Pi}$$

where the summation stands for summation over all points in the geometry. Replacing the integration over the cross section with the cross-sectional area (A) and the volume forces with line forces, the equation becomes

$$\delta W = \int_L (-\varepsilon_{n\text{test}} \sigma_n A + \mathbf{u}_{\text{test}}^t \mathbf{F}_L) dL + \sum_i \mathbf{u}_{\text{test}}^t \mathbf{F}_{Pi}$$

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 is 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

Introduction

You can use a [Spring-Damper](#) 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 interface 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.

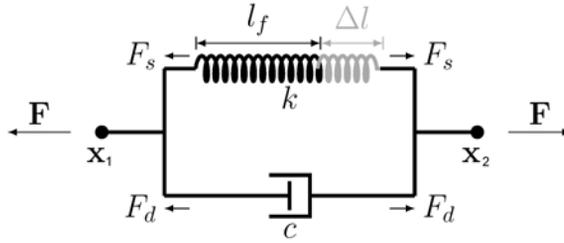


Figure 10-1: Conceptual sketch of a Spring-Damper.

Connection Points and Spring Length

The current position of the two endpoints, \mathbf{x}_1 and \mathbf{x}_2 can be written as

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{X}_1 + \mathbf{u}_1 \\ \mathbf{x}_2 &= \mathbf{X}_2 + \mathbf{u}_2\end{aligned}$$

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 Δl_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 Δl 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 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 η , 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

$$W_s = \frac{1}{4} i_{ac} \operatorname{Re}(\operatorname{lindev}(F_s) \cdot \operatorname{conj}(\operatorname{lindev}(\Delta l)))$$

$$W_d = \frac{1}{4} i_{ac} \operatorname{Re}(\operatorname{lindev}(F_d) \cdot \operatorname{conj}(\operatorname{lindev}(\Delta l)))$$

These energy quantities represent the cycle average, and only the perturbation terms are included.

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.



For a detailed overview of the functionality available in each product, visit <http://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}_{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.

DEPENDENT VARIABLES

The dependent variable (field variable) is for the **Displacement field u** which has two components (u, v) in 2D and three components ($u, v,$ and w) in 3D. The name can be changed but the names of fields and dependent variables must be unique within a model.

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.



- [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 are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).



In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

FEATURES AVAILABLE FROM SUBMENUS

Many features for the Truss interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases. The submenus at the **Edge** level (3D) or **Boundary** level (2D) are

- **Material Models**
- **Line and Volume Loads**
- **Mass, Spring, and Damper**
- **Line Constraints**

The submenus at the **Point** level are

- **Mass, Spring, and Damper**
- **More Constraints**
- **Pairs**

LINKS TO FEATURE NODE INFORMATION

These nodes (and subnodes) are described in this section (listed in alphabetical order):

- [Antisymmetry](#)
- [Cross Section Data](#)
- [Edge Load](#)
- [External Stress](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)
- [Linear Elastic Material](#)
- [Phase](#)
- [Pinned](#)
- [Point Mass](#)
- [Point Mass Damping](#)
- [Spring-Damper](#)
- [Straight Edge Constraint](#)
- [Symmetry](#)
- [Thermal Expansion \(for constraints\)](#)
- [Thermal Expansion \(for materials\)](#)

These nodes are described for the Solid Mechanics or Beam interface:

- Added Mass
- Damping
- Free
- Gravity¹
- Initial Values
- Plasticity
- Point Load
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rotating Frame¹
- Spring Foundation

¹ This is selected from the **Line and Volume Loads** submenu for this interface.



If there are subsequent constraints on the same geometrical entity, the last one takes precedence.



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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 enter the cross section area for the truss elements on the selected edges.



This is required input data.

BASIC SECTION PROPERTIES

Enter an **Area A**.

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.

ACTIVATION CONDITION

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 \(for materials\)](#)
- [Hygroscopic Swelling](#)
- [Initial Stress and Strain](#)
- [External Stress](#)
- [Damping](#)
- [Plasticity](#)

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** button () and select **Advanced Physics Options**.

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

Thermal Expansion (for materials)

Use the **Thermal Expansion** subnode to add an internal thermal strain caused by changes in temperature.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.

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 [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

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

where α is the secant coefficient of thermal expansion. α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. Use **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

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\text{hs}} = \beta_h(c_{\text{mo}} - c_{\text{mo,ref}})$$

where β_h is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{mo,ref}$ 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** σ_{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** drop-down list, stress tensors announced by any physics interface will be shown, and also the entry **User defined**. When **User defined** is selected, you can enter the data for the **External stress tensor** S_{ext} as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the global coordinate system, and are projected onto the tangential direction of the truss element. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type**—**Add to stress tensor** 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

Use a **Spring-Damper** 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** η .

FREE LENGTH

The free length is the distance between the connection points when there is no force in the spring. Select an option from the list—**Specify initial extension** or **Specify free length**.

- For **Specify initial extension** enter a value for Δl_0 . The free length is computed as $l_f = l_0 - \Delta l_0$, where l_0 is the initial distance between the connection points.
- For **Specify free length** enter a value for l_f .

ACTIVATION CONDITION

Select a **Spring action**—**Bidirectional**, **Tension only**, or **Compression only**.

Select the **Deactivation condition** check box to enter a **Deactivation indicator expression** i_{dac} . The expression is treated as a boolean expression, so that when it is evaluated to a nonzero value, the spring or damper is deactivated.

Select the **Permanently deactivate** 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

Ribbon

Physics tab with **Truss** selected:

Edges>Material Models>Spring-Damper

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** button () and select **Advanced Physics Options**.



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 α , the temperature T , and the strain-free reference temperature T_{ref} as

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

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of surroundings idealized by the constraints.

Select **Inherit from edge** to take the thermal expansion data from the domain being constrained. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual surrounding material has the same thermal expansion as the edge itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for **Temperature** T , specifying the temperature distribution of the surrounding material. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** α . As a default, values **From material** are used. This requires that a material has been assigned to the points where the constraint is active.
For **User defined** enter the coefficient of thermal expansion α .
- A value or expression for the **Volume reference temperature** T_{ref} which is the temperature at which there are no thermal displacements at the constraints.

Enter the coordinates of the **Reference point**, the point where the displacement is zero. The choice of reference point only affects the rigid body motion. If there are several different constraints with a **Thermal Expansion** subnode, the same reference point should usually be selected in all of them.



- [Constraints and Thermal Expansion](#) in the [Structural Mechanics Modeling](#) chapter.
 - [Thermal Expansion of Constraints](#) in the [Structural Mechanics Theory](#) chapter.
-

LOCATION IN USER INTERFACE

Context Menus

Truss>Pinned>Thermal Expansion

TrussPrescribed 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 drop-down 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** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)



[Symmetry Constraints](#)



You can activate and deactivate this boundary condition by assigning it to a constraint group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss>Line Constraints>Symmetry (Line)

Truss>More Constraints>Symmetry (Point)

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** button () and select **Advanced Physics Options**.



In the *COMSOL Multiphysics Reference Manual*:

- [Constraint Reaction Terms](#)
- [Weak Constraints](#)
- [Constraint Settings](#)



[Symmetry Constraints](#)



You can activate and deactivate this boundary condition by assigning it to a constraint group. See [Load Cases](#) in the [Structural Mechanics Modeling](#) chapter.

LOCATION IN USER INTERFACE

Context Menus

Truss>Line Constraints>Antisymmetry (Line)

Truss>More Constraints>Antisymmetry (Point)

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.

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 ³	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** α_{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 (ϕ), which is specified in the **Phase** subnode.

Together these define a harmonic load, for which the amplitude and phase shift can vary with the excitation frequency, f :

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

PHASE

Add the phase load F_{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

Multiphysics Interfaces and Couplings

The Structural Mechanics Module contains predefined multiphysics interfaces to facilitate easy set up 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 contain 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 Interface](#)
- [The Fluid-Solid Interaction Interface](#)
- [The Fluid-Shell Interaction Interface](#)
- [The Fluid-Membrane Interaction Interface](#)
- [The Fluid-Solid Interaction, Fixed Geometry Interface](#)
- [The Fluid-Shell Interaction, Fixed Geometry Interface](#)
- [The Fluid-Membrane Interaction, Fixed Geometry Interface](#)
- [The Joule Heating and Thermal Expansion Interface](#)

- The Joule Heating and Thermal Expansion Interface
- The Piezoelectric Devices Interface
- The Magnetostriction Interface
- Hygroscopic Swelling Coupling
- Couplings Between Structural Mechanics Interfaces
- Multiphysics Couplings
 - Thermal Expansion (Multiphysics Coupling)
 - Thermal Expansion, Layered Shell (Multiphysics Coupling)
 - Hygroscopic Swelling (Multiphysics Coupling)
 - Piezoelectric Effect
 - Magnetostriction
 - Fluid-Structure Interaction
 - Fluid-Structure Interaction, Pair
 - Fluid-Structure Interaction, Fixed Geometry
 - Solid-Shell Connection
 - Solid-Beam Connection
 - Shell-Beam Connection

The Thermal Stress Interface

The **Thermal Stress** () 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** 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 Couplings** node is added, which automatically includes the multiphysics coupling features **Thermal Expansion** and **Temperature Coupling**.

On the Constituent Physics Interfaces

The Solid Mechanics interface is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. In 2D, plane stress or plane strain assumptions can be used. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.

The Heat Transfer in Solids interface provides features for modeling heat transfer by conduction, convection, and radiation. A Heat Transfer in Solids model is active by default on all domains. All functionality for including other domain types, such as a fluid domain, is also available. The temperature equation defined in solid domains corresponds to the differential form of the Fourier's law that may contain additional contributions like heat sources.



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 Approaches](#) 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**, 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 Couplings** 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 Couplings** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 11-1: MODIFIED SETTINGS FOR A THERMAL STRESS INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior the Structural transient behavior is set to Quasi-static .
Heat Transfer in Solids	For the Heat Transfer in Solids interface, under Discretization, the shape function order for the temperature is set to Linear .
Thermal Expansion	The Domain Selection is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Heat Transfer in Solids interfaces are preselected in the Thermal Expansion section.
Temperature Coupling	The corresponding Solid Mechanics and Heat Transfer in Solids interfaces are preselected in the Coupled Interfaces section (described in the <i>COMSOL Multiphysics Reference Manual</i>).

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 [The Fluid-Solid Interaction Interface](#) is used to couple the interfaces.
- The [Temperature Coupling](#) feature node is described for [The Joule Heating Interface](#) in the *COMSOL Multiphysics Reference Manual*.



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 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 coupled fluid-structure interaction.

You add a predefined **Fluid-Solid Interaction** interface from the **Fluid-Structure Interaction** () group in 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.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see [Deforming Fluid Domains](#).

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

- The Laminar Flow interface solves the Navier-Stokes equations for the conservation of momentum and mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.



If you also have the CFD Module, you can add a **Stationary, One-Way Coupled with Initialization** or **Transient, One-Way Coupled with Initialization** study for turbulence models requiring the wall distance. See [Stationary, One-Way Coupled with Initialization](#) and [Time Dependent, One-Way Coupled with Initialization](#) in the *COMSOL Multiphysics Reference Manual* for more information.

- The Solid Mechanics interface is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.



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 Approaches](#) 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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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 coupled fluid-structure interaction.

You add a predefined **Fluid-Shell Interaction** interface from the **Fluid-Structure Interaction** () group in the **Fluid Flow** branch () of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Shell** interfaces are then added to the Model Builder.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see [Deforming Fluid Domains](#).

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

- The Laminar Flow interface solves the Navier-Stokes equations for the conservation of momentum and mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.



If you also have the CFD Module, you can add a **Stationary, One-Way Coupled with Initialization** or **Transient, One-Way Coupled with Initialization** study for turbulence models requiring the wall distance. See [Stationary, One-Way Coupled with Initialization](#) and [Time Dependent, One-Way Coupled with Initialization](#) in the *COMSOL Multiphysics Reference Manual* for more information.

- The Shell interface is used to model structural shells on boundaries in 3D and 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. Geometric nonlinearity can be taken into account. The material is assumed to be linearly elastic.

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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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 coupled fluid-structure interaction.

You add a predefined **Fluid-Membrane Interaction** interface from the **Fluid-Structure Interaction** () group in the **Fluid Flow** branch () of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Membrane** interfaces are then added to the Model Builder.

At the same time, a **Moving Mesh** node with a **Deforming Domain** subnode is created under **Definitions**. For more information, see [Deforming Fluid Domains](#).

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

- The Laminar Flow interface solves the Navier-Stokes equations for the conservation of momentum and mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.



If you also have the CFD Module, you can add a **Stationary, One-Way Coupled with Initialization** or **Transient, One-Way Coupled with Initialization** study for turbulence models requiring the wall distance. See [Stationary, One-Way Coupled with Initialization](#) and [Time Dependent, One-Way Coupled with Initialization](#) in the *COMSOL Multiphysics Reference Manual* for more information.

- The Membrane interface is mainly used to model prestressed 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. When a membrane is used by itself, a tensile prestress is necessary in order to avoid singularity because a membrane with no stress or compressive stress has no transverse stiffness. The material model is linearly elastic or hyperelastic.

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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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-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 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 Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

The Laminar Flow interface solves the Navier-Stokes equations, for conservation of momentum, and the continuity equation, for conservation of mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.

The Solid Mechanics interface is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. In 2D, plane stress or plane strain assumptions can be used. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.

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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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, 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 the **Fluid Flow** () of the **Model Wizard** or **Add Physics** windows. **Laminar Flow** and **Shell** interfaces are then added to the Model Builder.

In addition, the **Multiphysics Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

The Laminar Flow interface solves the Navier-Stokes equations, for conservation of momentum, and the continuity equation, for conservation of mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.

The Shell interface is used to model structural shells on boundaries in 3D and 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. Geometric nonlinearity can be taken into account. The material is assumed to be linearly elastic.

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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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, 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 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 Couplings** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction**.

On the Constituent Physics Interfaces

The Laminar Flow interface solves the Navier-Stokes equations, for conservation of momentum, and the continuity equation, for conservation of mass, to compute the fluid velocity and pressure. The interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). When the CFD Module is available, turbulent flow is supported through the inclusion of a number of turbulence models.

The Membrane interface is mainly used to model prestressed 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. When a membrane is used by itself, a tensile prestress is necessary in order to avoid singularity because a membrane with no stress or compressive stress has no transverse stiffness. The material model is linearly elastic or hyperelastic.

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 Couplings** node. You can choose from the available coupling features, but the modified settings are not included.

Another difference is that the Laminar Flow interface added by the predefined coupling employs pseudo-time stepping per default for stationary models, while the standalone version uses an automatic Newton damping. The latter can sometimes give shorter solution times, while the pseudo-time stepping often gives a more robust solution procedure. You can disable the pseudo-time stepping in **Advanced Settings** section for the fluid interface.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 (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.

- 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 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 Couplings** node is added, which automatically includes the multiphysics coupling features **Thermal**, **Expansion Electromagnetic Heat Source**, **Boundary Electromagnetic Heat Source**, and two **Temperature Coupling** nodes.

On the Constituent Physics Interfaces

The Electric Currents interface computes electric field, current and potential distributions in conducting media under conditions where inductive effects are negligible; that is, when the skin depth is much larger than the studied device. Depending on the licensed products, time and frequency domain formulations that account for capacitive effects are also provided. The Electric Currents interface solves a current conservation equation based on Ohm's law using the scalar electric potential as the dependent variable.

The Solid Mechanics interface is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. In 2D, plane stress or plane strain assumptions can be used. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.

The Heat Transfer in Solids interface provides features for modeling heat transfer by conduction, convection, and radiation. A Heat Transfer in Solids model is active by

default on all domains. All functionality for including other domain types, such as a fluid domain, is also available. The temperature equation defined in solid domains corresponds to the differential form of the Fourier's law that may contain additional contributions like heat sources.



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 Approaches](#) 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 Couplings** node. You can then choose from the available coupling features, **Thermal Expansion**, **Electromagnetic Heating**, and **Temperature Coupling**, but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics Couplings** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 11-2: MODIFIED SETTINGS FOR A JOULE HEATING AND THERMAL EXPANSION INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Solid Mechanics	For the Solid Mechanics interface, under Structural Transient Behavior, the Structural transient behavior is set to Quasi-static .
Electric Currents	No changes.

TABLE 11-2: MODIFIED SETTINGS FOR A JOULE HEATING AND THERMAL EXPANSION INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
Heat Transfer in Solids	For the Heat Transfer in Solids interface, under Discretization, the shape function order for the temperature is set to Linear .
Thermal Expansion	The Domain Selection is the same as that of the participating physics interfaces. The corresponding Solid Mechanics and Heat Transfer in Solids interfaces are preselected in the Thermal Expansion section.
Electromagnetic Heating	The Domain Selection is the same as that of the participating physics interfaces. The corresponding Electric Currents and Heat Transfer in Solids interfaces are preselected in the Electromagnetic Heating section.
Temperature Coupling	Two Temperature Coupling nodes (described in the <i>COMSOL Multiphysics Reference Manual</i>) are generated, one for the transfer of temperatures to the Solid Mechanics interface and one for the transfer of temperatures to the Electric Currents interface.

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 [The Fluid-Solid Interaction Interface](#) coupling feature node is described for [The Thermal Stress Interface](#).
- The [Electromagnetic Heating](#), and [Temperature Coupling](#) coupling feature nodes are 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 (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.

- 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.

The Piezoelectric Devices Interface

The **Piezoelectric Devices** interface () combines Solid Mechanics and Electrostatics together with the constitutive relationships required to model piezoelectrics. 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 **Piezoelectric Devices** 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 Couplings** 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 Piezoelectric Devices 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 intended for general structural analysis of 3D, 2D, or axisymmetric bodies. In 2D, plane stress or plane strain assumptions can be used. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.

The Electrostatics interface is used to compute the electric field, the electric displacement field and potential distributions in dielectrics under conditions where the electric charge distribution is explicitly prescribed. The formulation is stationary but for use together with other physics interfaces, also eigenfrequency, frequency-domain, small-signal analysis and time-domain modeling are supported in all space dimensions.

The physics interface solves Gauss' law for the electric field using the scalar electric potential as the dependent variable.



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 Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Piezoelectric Devices**, 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 Couplings** 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 Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 Piezoelectric Devices 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.



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 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.

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 Couplings** 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 non isotropic magnetostrictive materials.

On the Constituent Physics Interfaces

The Solid Mechanics interface is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. The Solid Mechanics interface is based on solving Navier’s equations, and results such as displacements, stresses, and strains are computed.

The Magnetic Fields interface solves the Ampère’s law for the magnetic flux density field by using the vector magnetic potential as the dependent variable.

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 Couplings** 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 Couplings** node) or from the **Physics** toolbar, **Multiphysics** 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 \(Multiphysics Coupling\)](#).

Couplings Between Structural Mechanics Interfaces

Some of the structural mechanics interfaces are formulated using different types of degrees of freedom. This needs to be taken into account in models where such interfaces interact. You can set up common transitions using the predefined multiphysics couplings. These are:

- [Solid-Shell Connection](#)
- [Solid-Beam Connection](#)
- [Shell-Beam Connection](#)

Multiphysics Couplings

Thermal Expansion (Multiphysics Coupling)

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 default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (`_`) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `tel`.

DOMAIN SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains to define the coefficient of thermal expansion and the different temperatures that cause thermal stress, or select **All domains** as needed.

When **Thermal Expansion** is added as an effect of adding a Thermal Stress interface, the selection is the same as for the participating physics interfaces.

Only domains that are active in the physics interfaces selected in the [Coupled Interfaces](#) section can be selected.

MODEL INPUTS

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you

to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.



See also [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

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 α is the secant coefficient of thermal expansion. α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\epsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

HEAT SOURCES

Select **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** button () and select **Advanced Physics Options**.

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 (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when 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.



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 Approaches](#) in the *COMSOL Multiphysics Reference Manual*.



- [Thermoelastic Damping](#)
- [Entropy and Thermoelasticity](#)

Thermal Expansion, Layered Shell (Multiphysics Coupling)

Use the **Thermal Expansion, Layered Shell** multiphysics coupling () to add an internal thermal strain caused by changes in temperature and account for the corresponding mechanical losses in the heat balance in layered materials represented by boundaries.

The **Thermal Expansion, Layered Shell** node is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>).

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern <name>.<variable_name>. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the name string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `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 **Layered Shell** lists include all applicable physics interfaces. The first physics interface of each type in the component is selected as the default.

You can also select **None** from either list to uncouple the **Thermal Expansion, Layered Shell** 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.

LAYER SELECTION

Select the applicable layers (the default setting is **All layered materials**) 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, next to the **Layered Material** list).

When a layered material stack or link is selected from the **Layered Material** list, deselect 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 Layer Selection Section](#).



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 Material](#) in the *COMSOL Multiphysics Reference Manual* for details on the definition of layered materials.



See [Thermal Expansion \(Multiphysics Coupling\)](#) for a description of the corresponding multiphysics coupling in domains.

MODEL INPUT

The **Volume reference temperature** T_{ref} is the temperature at which there are no thermal strains. As a default, the value is obtained from a **Common model input**. You can also select **User defined** to enter a value or expression for the temperature locally.

When using **Common model input**, you can see or modify the value of the volume reference temperature by clicking the **Go To Source** button (). This will move you to the **Common Model Inputs** node under **Global Definitions** in the Model Builder. The default value is room temperature, 293.15 K.

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** in the current component, having the same selection as the current **Thermal Expansion** node.



See also [Common Model Inputs](#) and [Model Input](#) in the *COMSOL Multiphysics Reference Guide*.

THERMAL EXPANSION PROPERTIES

Select an **Input type** to select how the thermal strain is specified. The default is **Secant coefficient of thermal expansion**, in which case the thermal strain is given by

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

where α is the secant coefficient of thermal expansion. α can be temperature dependent.

When **Input type** is **Tangent coefficient of thermal expansion**, the thermal strain is given by

$$\varepsilon_{\text{th}} = \exp\left(\int_{T_{\text{ref}}}^T \alpha_t(\tau) d\tau\right) - 1$$

where α_t is the tangential coefficient of thermal expansion.

When **Input type** is **Thermal strain**, enter the thermal strain dL as function of temperature explicitly.

In all three cases, the default is to take values **From material**. When entering data as **User defined**, select **Isotropic**, **Diagonal** or **Symmetric** to enter one or more components for a general coefficient of the thermal expansion tensor or the thermal strain tensor. When a nonisotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

HEAT SOURCES

Select **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** button () and select **Advanced Physics Options**.



- [Thermoelastic Damping](#)
- [Entropy and Thermoelasticity](#)

Hygroscopic Swelling (Multiphysics Coupling)

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 β_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 default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the name string must be unique. Only letters, numbers, and underscores (`_`) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `hs1`.

HYGROSCOPIC SWELLING PROPERTIES

Enter a **Strain reference concentration** $c_{\text{mo,ref}}$. This is the concentration at which there are no strains due to hygroscopic swelling.

Enter the **Molar mass** of the fluid, M_{m} . The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** from the list to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h} . The default value for the **User defined** case is $1.5\text{e-}4 \text{ m}^3/\text{kg}$. 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.

HYGROSCOPIC SWELLING

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 Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

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 requires no user settings when it is coupled with the multiphysics coupling node.

SETTINGS

The **Label** is the default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (`_`) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `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.

PIEZOELECTRIC EFFECT

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 (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a Piezoelectric Devices 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 Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

Magnetostriction

The **Magnetostriction** multiphysics coupling 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.

COUPLING TYPE

From the list, choose one of these coupling types:

- **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.
- **Fully coupled** (the default) to include both the direct and inverse magnetostrictive effects.

In 2D and 2D axial symmetry geometries only, select the **Lorentz force contribution** check box to include the electric field changes and the body load caused by the material motion in a magnetic field. This has no effect under stationary studies.



- [Velocity \(Lorentz Term\)](#)

The **Fluid-Structure Interaction** () multiphysics coupling 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 Shell or Membrane interfaces is used.

The effect of the coupling depends on whether an **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 default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the name string must be unique. Only letters, numbers, and underscores (`_`) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `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 **Solid** 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 (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **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 **Solid** 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 Approaches](#) 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 coupling 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 union when the boundary is shared, use the [Fluid-Structure Interaction](#) multiphysics coupling.

The effect of the coupling depends on whether an **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 default multiphysics coupling feature name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics

interfaces, the name string must be unique. Only letters, numbers, and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first multiphysics coupling feature in the model) is `fsi1`.

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 **Solid** 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 (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when the physics interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating physics interfaces are selected.

You can also select **None** from either list to uncouple the **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 **Solid** 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 Approaches](#) 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 coupling 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.

Solid-Shell Connection

Add the **Solid-Shell Connection** multiphysics coupling node () to create transitions between domains modeled using the Solid Mechanics or Multibody Dynamics interfaces and boundaries modeled using the Shell interface.

The **Solid-Shell Connection** node is only available with some COMSOL products (see <http://www.comsol.com/products/specifications/>)

CONNECTION SETTINGS

Select the **Connection type** — **Solid boundaries to shell edges**, **Solid and shell shared boundaries**, or **Solid and shell parallel boundaries**.

For the two first 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 underconstrained if the mesh on the solid is very coarse.

You can modify the **Connection tolerance Δ** when either of the options **Shell thickness** or **Distance from shell midsurface** are used. The search distance is slightly extended by the distance Δ . The default value is 0.5% of the shell thickness, which allows for small inaccuracies on for example a curved geometry.

Solid and Shell 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.

Solid and Shell 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**, **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 **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.



- 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/Tutorials/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.

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 underconstrained if the mesh on the solid is very coarse.

If needed, modify the **Connection tolerance Δ** . The search distance is slightly extended by the distance Δ . The default value is 0.5% of the beam height.

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.

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.

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.

This coupling introduces a set of extra degrees of freedom, which preferably should 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 Boundary \(3D\)](#) 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 coupling 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 <http://www.comsol.com/products/specifications/>)

CONNECTION SETTINGS

Select the **Connection type** — **Shell edges to beam points**, **Shell and beam shared edges**, **Shell and beam parallel edges**, or **Shell boundaries to beam points**.

For all options except **Shell and beam 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.

Shell and Beam 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** ζ . For **Offset vector** enter values for \mathbf{d}_0 in the table. The offset is the vector from the reference surface of the shell to the actual position of the beam. The **Offset vector** is interpreted in the coordinate system selected in the **Coordinate System Selection** section.

Shell and 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

values. 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/Tutorials/shell_beam_connection**

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.

quasi-static 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 effective 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 856
- absolute values 376
- absolute-tolerance parameters 242
- abstract rigid domain 426
- acceleration loads 86
- acoustic-structure interaction, frequency
 - domain interface 196
- activation 190
- activation (node) 607
- added mass (node) 680
 - layered shell interface 914
 - shell and plate interfaces 835
- added mass, interface (node) 915
- added mass, theory 483
- adding
 - connections 75
- additive strain decomposition (check box) 544, 551, 556, 564
- adhesion (node) 718
- alpha coefficient 590
- Anand viscoplasticity 398
- angle of internal friction 349, 357, 591
- angular acceleration and velocity 422
- anisotropic materials
 - defining 293
 - elastic properties 544, 935
- antisymmetry (node)
 - beam interface 1005
 - layered shell interface 901
 - shell and plate interfaces 822
 - solid mechanics 656
 - truss interface 1074
- Application Libraries window 33
- application library examples
 - beam cross sections 1022
 - beams 978
 - Cam-Clay material 554
 - circumferential mode extension 532
 - creep models 582
 - cross section data 981
 - damping 617
 - eigenfrequency analysis 44
 - fixed constraint 646
 - geometric nonlinearity 167
 - harmonic perturbation 56
 - initial stress and strain 599, 601, 937
 - large plastic strains 579
 - layered shells 872
 - linear buckling study 62
 - linear elastic material 989
 - load cases 83–84
 - membrane interface 931
 - MEMS materials database 108
 - modal mass 44, 182, 207
 - model superposition 51, 53
 - Mooney-Rivlin, two parameters 561
 - Ogden 561
 - piezoelectric devices 1107, 1111
 - piezoelectric materials database 108
 - plasticity models 579
 - prestressed bolts 63, 222
 - response spectrum analysis 64, 232
 - rigid connector 695
 - sharing edges 78, 1130, 1136
 - shells 782
 - soil plasticity 592
 - solid mechanics 534
 - spring foundation 194
 - thermal expansion 596
 - thermal stress 1083
 - trusses 1059
 - viscoelastic material 574

- viscoplasticity models 584
- applied force (node) 630, 698
 - theory 428
- applied moment (node) 632, 700
 - theory 428
- applying
 - loads 81
 - moments 85
- Arbitrary Lagrangian-Eulerian (ALE)
 - method 162
- area scale factor, layered shell 860
- area, beam cross sections 1026
- Arruda-Boyce material 332
- associated flow rule 368
- attachment (node)
 - layered shell interface 919
 - shells 850, 1016
 - solid mechanics 704
 - theory 427
- augmented Lagrangian method 25, 173, 498
- axial stress 1041
- axial symmetry
 - constraints and 92
 - initial stress and strain 280
- axisymmetric models
 - solid mechanics 67
- azimuthal mode number 272
- azimuthal wave number 489
- B**
 - back stress 364, 400
 - beam
 - coupling to a solid 76–77
 - beam connection (node) 1011
 - shells 844
 - solid mechanics 707
 - beam cross section interface 1043
 - theory 1026
 - beam interface 975
 - theory 942
 - beams
 - cross section data 980
 - initial loads and strains 953
 - initial stresses and strains 992
 - linear elastic material 988
 - loads applied 1009
 - prescribed acceleration 998
 - prescribed displacement/rotation 994
 - prescribed velocity 997
 - section orientation 985
 - strain-displacement/rotation 951
 - stress evaluation 956
 - thermal expansion 990
 - thermal strain 952
 - Beltrami-Michell 1032
 - bending shear stress 1041
 - bending shear stresses 1028
 - bending stress 1041
 - biaxial compression 373
 - biaxial data 375
 - biaxial tension 593
 - bimoment 1040
 - Blatz-Ko material 334
 - body load (node)
 - layered shell interface 903
 - shell and plate interfaces 823
 - solid mechanics 659
 - bolt pretension (node) 721
 - bolt pre-tension, theory 217
 - bolt selection (node) 722
 - bolt thread contact (node) 724
 - bolted joints 217
 - boundary conditions
 - contact pairs 177
 - shell and plate interfaces 745
 - boundary load (node) 664
 - layered shell interface 906

- boundary nodes
 - beam interface 978
 - layered shell 872
 - membrane interface 931
 - shell and plate interfaces 783
 - solid mechanics 534
 - truss interface 1059
- boundary pair (node) 687
- box sections, beams 981
- Bresler-Pister criterion 372
- buckling 24
- built-in couplings 75
- bulk modulus
 - elastic moduli 290, 542, 672
- Burgers vector 397
- C** calcite 376
- calculating stress and strain 747
- Cam-clay model 378
- canonical systems 251
- carbonate rocks 376
- Cauchy stress 274
- Cauchy stress tensor 160
- Cayley-Hamilton theorem 261, 277
- cell periodicity 687
- cell periodicity (node)
 - solid mechanics 684
- center of gravity, beam cross sections
 - 1026
- center of mass
 - boundary (node) 634
 - edge (node) 634
 - point (node) 635
 - rigid domains 422
- center of rotation 1030
 - boundary (node) 696
 - edge (node) 696
 - point (node) 696
 - rigid domains 424
- centrifugal acceleration loads 86
- ceramics 373
- change thickness (node)
 - shell and plate interfaces 786
 - solid mechanics 539
- charge conservation, piezoelectric
 - (node) 565
- circle, Mohr 349
- circular sections, beams 981
- circumferential mode number 272
- Coble creep 397
- coefficient of thermal expansion
 - beams 990
 - shells and plates 794
- cohesion 349, 591
- cohesion sliding resistance 501
- cohesionless soils 356
- cohesive-frictional materials 373
- common sections, beam interface 981
- common settings 30
- complementarity 368
- complex mechanical energy flux 516
- complex modulus 153
- compressive meridians 277, 350, 373–374
- concrete 372, 374
- concrete (node) 592
- conduction loss (time-harmonic) (node)
 - 620
- connection
 - beam edge to shell edge 760
 - beam perpendicular to solid 758
 - beam point to shell boundary 761
 - beam point to shell edge 762
- connections
 - beams, shells 759
 - beams, shells, solids 752
 - shell perpendicular to solid 753
- consistency parameter 368

- constitutive relation, membranes 927
- constraints 89
- contact (node) 710
- contact formulation 182
- contact help variables 501
- contact modeling, friction 502
- continuity (node)
 - layered shell interface 921
- continuity, layered shell 865
- converse piezoelectric effect 122
- coordinate system, beam cross sections 1028
- coordinate systems
 - constraints and 90
 - loads and 81
 - local edge system 812
 - solid mechanics theory 251
- coordinate systems, membranes 926
- Coulomb friction 716
- coupling
 - beam to a shell 78
 - beam to a solid 76–77
 - shell to a solid 76
- coupling loss (node) 618
- coupling operator 78
- creep (node) 579
 - layered shell interface 879
- creep dissipation rate density 401
- creep strain rate 389
- critical load factor 61
- cross section (node) 980
- cross section data (node) 1061
- crystal cleavage 376
- crystal cut standards 125
- crystalline solids and creep 391
- curvature 953
- cyclic symmetry, theory 488
- cylindrical coordinate systems 67

D

- damage (node) 605
- damped eigenfrequency study 152
- damping
 - equation of motion, and 147
 - loss factors 151
 - losses and 146
 - point mass 838, 1010, 1077
 - solid mechanics 614
 - viscoelastic materials, and 317
- damping (node)
 - layered shell interface 886
 - solid mechanics 613, 801
- damping models 151
- decohesion 504
- defining
 - anisotropic materials 293
 - constraints 89
 - isotropic materials 289
 - multiphysics models 195
 - orthotropic materials 292
 - thermoelastic materials 295
- deformation gradient 255
- deformation resistance 399
 - saturation coefficient 399
 - sensitivity 399
- destinations and sources 174
- deviatoric creep 393, 396
- deviatoric stress 343
- dielectric loss (node) 619
- dielectric loss factor 134
- diffusional creep 397
- dilatational contributions 321
- direct piezoelectric effect 122, 407
- discretization, layered shell 861
- dislocation creep 398
- dispersion curves 47
- displacement field, defining 156
- displacement gradients 254

- displacement variables
 - element types and 74
- dissipated energy 515
- dissipated energy density 371, 401
- dissipated energy density rate 312
- dissipation, piezoelectric materials 410
- distortional contributions 321
- disturbance factor 377
- documentation 32
- dolomite 376
- domain nodes
 - solid mechanics 534
- double dogleg nonlinear solver 184
- Drucker-Prager criterion 351
- ductile materials 345
- Dulong-Petit law 295
- dynamic cyclic symmetry 488
- dynamic frictional coefficients 503
- E**
 - edge load (node)
 - beam interface 1006
 - layered shell interface 907
 - shell and plate interfaces 826
 - solid mechanics 666
 - truss interface 1075
 - edge nodes
 - beam interface 978
 - membrane interface 931
 - solid mechanics 534
 - truss interface 1059
 - effective creep strain rate 393
 - effective plastic strain rate 342
 - effective stress 1042
 - effective stress tensor 377
 - eigenfrequency study 42
 - solid mechanics 449
 - eigenvalue solvers 43
 - eight-chain model 333
 - elastic deformation tensor 264
 - elastic energy 513
 - elastic Green-Lagrange strain tensor 265
 - elastic material properties 542–544, 934–935
 - elastic moduli 289
 - elastic right Cauchy-Green tensor 264
 - elastic strain energy 337
 - elastic volumetric strain variable 381
 - elasticity matrix 289
 - elastoplastic materials 341
 - elcontact variable 499
 - element types 74
 - elliptic cap 353
 - elplastic 368, 370
 - emailing COMSOL 34
 - energy dissipation 515
 - energy function 319
 - energy quantities 513
 - equation of motion, damping and 147
 - equivalent single layer shell theory 764
 - equivalent viscous damping 152
 - evanescent modes 46
 - excitation frequency 152
 - explicit damping 153
 - external loads, shell and plate interfaces 747
 - external strain (node) 602, 604
 - layered shell interface 885
 - external stress (node) 600, 798, 938, 993, 1067
 - layered shell interface 884
 - external stress-strain relation (node) 621
- F**
 - face load (node) 824
 - layered shell interface 904
 - failure surfaces 341
 - Filter 192
 - first order shear deformation laminate theory 766

- first Piola-Kirchhoff stress 274
- fixed constraint (node) 627, 646
 - layered shell interface 896
 - shell and plate interfaces 802
- fixed constraint, interface (node) 897
- fixed constraint, theory 427
- Floquet periodicity, theory 488
- flow rule 367
- fluid pore pressure 355
- fluid-membrane interaction, fixed geometry 1098
- fluid-shell interaction 1087
- fluid-solid interaction 1084
- fluid-ssolid interaction 1090
- fluid-structure interaction 1093, 1124, 1128
- fluid-structure interaction interface theory 494
- fluid-structure interaction, fixed geometry 1095
- fluid-structure interaction, pair 1126
- fold lines 743
- foldline connection, layered shell 864
- fold-line limit angle 780, 870
- follower loads 928
- force linear strains (check box) 544, 550, 555, 564, 567, 570, 622, 936
- free (node) 637
 - layered shell interface 888
- free-free modes 45
- frequency domain study
 - solid mechanics 448
- frequency domain, prestressed study 167
- frequency response study
 - loss factor damping 152
- friction (node) 716
- friction forces 717
- friction in contact modeling 502
- friction models 716
- friction traction penalty factor 501
- friction, angle 349, 357, 591
- frictionless materials 350, 352

G

- Gao material 335
- gap distance variable 500
- Garofalo law 396
- Gauss points 368, 370
- general extrusion operator 78
- generalized Hoek-Brown criterion 376
- generalized Maxwell model 310
- Gent material 333
- geological strength index (GSI) 376
- geometric nonlinearity 163
 - membranes and 924
 - micromechanics, and 155
 - piezoelectric devices 167
- glass transition temperature 315
- global
 - coordinate systems 251
- GMRES iterative solvers 240
- gradient displacements 254
- gravity (node) 660
 - layered shell interface 905
- Green's theorem 1035
- Green-Lagrange strain 156
- Green-Lagrange tensor 257

H

- Haigh–Westergaard coordinates 278
- hardening constant 399
- hardening models
 - theory 360
- hardening sensitivity 399
- harmonic loads 448
- harmonic perturbation 54
- harmonic perturbation (node)
 - truss interface 728, 853, 1017
- hear center 1030
- hear correction factor 1029

- heat dissipation 153
- Hencky plastic strain 370
- Hermitian matrices 240
- hexagonal prism 344
- Hill orthotropic plasticity 358
- Hill's effective stress 360
- Hoek-Brown criterion 375
- hole (node) 1044
- H-profile sections, beams 981
- hydrostatic axis 278
- hydrostatic pressure 275, 349
- hydrostatic stress 356
- hygroscopic swelling 87
- hygroscopic swelling (node) 597
 - layered shell interface 882
- hygroscopic swelling (node), multiphysics 1120
- hyperbolic sinus 396
- hyperelastic material 514
- hyperelastic material (node) 556
- hyperelastic materials 326
 - nearly incompressible 323
 - theory 319
- hysteretic loss 131
- I**
 - I.R.E. standard, for material orientation 123
 - ideal plasticity 361
 - IEEE standard, for material orientation 123
 - imperfection sensitivity 62
 - implementation
 - beams 954
 - trusses 1052
 - include geometric nonlinearity (checkbox) 545
 - inelastic deformation tensor 265
 - inertial effects, contact modeling 182
 - inertial forces 423
 - initial loads and strains, beams 953
 - initial stress and strain 412
 - theory 279
 - initial stress and strain (node) 598
 - beam interface 992
 - layered shell interface 883
 - membrane interface 936
 - shell and plate interfaces 796
 - solid mechanics 598
 - truss interface 1066
 - initial values (node)
 - beam interface 980
 - layered shell interface 875
 - rigid domains 625
 - rigid domains, theory 424
 - shell and plate interfaces 785
 - solid mechanics 538
 - initial yield stress 361–362
 - integration in a laminate 766
 - internet resources 32
 - invariants 319
 - inverse piezoelectric effect 407
 - isochoric
 - contributions 321
 - process 255
 - strain energy density 323
 - isochoric strain energy 337
 - isochoric-elastic
 - deformation 322
 - Green-Lagrange strain tensor 322
 - right Cauchy-Green tensor 322
 - isotropic hardening 361
 - isotropic materials
 - defining 289
 - elastic properties 542, 934
 - isotropic plasticity 341
 - isotropic rocks 376
 - iterative solvers 240

- J**
 - joints, bolted 217
 - Joule heating and thermal expansion interface 196, 1101
- K**
 - k coefficient 592
 - Kelvin–Voigt viscoelastic model 313
 - kinematic constraints 97
 - kinematic hardening 363
 - kinematics, rigid domain 420
 - kinetic energy 515
 - knowledge base, COMSOL 35
 - Kuhn-Tucker conditions 368
- L**
 - Lade-Duncan criterion 356
 - Lagrange elements 71
 - Lagrange shape functions, trusses and 1050
 - Lagrangian formulations 253
 - Lamé parameters 290, 542
 - lamina constitutive law 767
 - laminated constitutive law 770
 - laminated constitutive law 770
 - large deformation modeling 253
 - large deformations 24
 - piezoelectric materials 167
 - large plastic strain 367
 - large strain plasticity 321
 - layered linear elastic material (node) 789
 - layered shell interface 869
 - theory 856
 - left Cauchy-Green tensor 258
 - Lie derivative 369
 - limestone 376
 - limiting chain extensibility 334
 - line load (node) 908
 - linear buckling study 24, 167
 - linear elastic material (node) 539, 792, 933
 - beam interface 988
 - layered shell interface 876
 - shell and plate interfaces 788
 - truss interface 1063
 - linear elastic materials 105
 - linear viscoelasticity 308
 - linearized buckling analysis 61
 - linper operator 54
 - load cases 83
 - load multiplier 61
 - loads
 - acceleration 86
 - applied to beams 1009
 - pressure 86
 - singular 84
 - total 87
 - local coordinate systems 252
 - local edge system 812
 - location
 - boundary (node) 632
 - edge (node) 632
 - point (node) 632
 - locking 325
 - Lode angle 277
 - Hoek-Brown criterion 376
 - Mohr-Coulomb criterion 350
 - Tresca criterion 344
 - Willam-Warnke criterion 373
 - logarithmic decrement 147
 - logarithmic plastic strain 370
 - long-term shear modulus 310
 - loss factor damping
 - solid mechanics and 151
 - solid mechanics theory 151
 - springs, and 194
 - loss modulus 153, 317
 - losses and damping 146
 - low-reflecting boundary (node) 688
 - low-reflecting boundary, theory 487
- M**
 - macroscopic shear modulus 333
 - magnetostriction 412

- magnetostriction (node) 1123
- magnetostriction multiphysics interface 1109
- magnetostrictive material 568
- magnetostrictive material (node) 568
- marble 376
- mass and moment of inertia (node) 633, 701
 - theory 426
- mass density 255
- mass matrix scaling 58
- mass moment of inertia 1009
- material coordinates 250
- material frame 156
- material models 103
- materials
 - nearly incompressible 106, 296
 - piezoelectric 407
 - viscoelastic 308
- materials, hyperelastic 319
- Matsuoka-Nakai criterion 355
- max scaling 58
- max shear stress factor 1029
- mechanical damping (node) 617
- mechanical energy flux 516
- membrane interface 929
 - theory 924
- meridians, tensile and compressive 277, 350, 373–374
- metal plasticity 342
- metals 345, 351
- metals and creep 391
- Mindlin plate theory 753
- Mindlin-Reissner type shell 737
- MITC shell formulation 737, 742
- mixed formulations 107, 297
- mobilized planes 355
- mode analysis study 46
- modeling, large deformations 253
- modified Cam-clay model 378
- modified Mohr-Coulomb criterion 358
- modified tensors 322
- Mohr's circle 1027
- Mohr-Coulomb criterion 349
- moment computations 99
- moments 423
 - beams 984
 - shells and plates 747
 - solid mechanics and 85
- moments of inertia 942, 1020
- moments of inertia, beam cross sections 1026
- Mooney-Rivlin material
 - five parameters 328
 - nine parameters 328
 - two parameters 327
- moving mesh interface, piezoelectric devices and 170
- MPH-files 33
- mu coefficient 591
- Mukherjee-Bird-Dorn equation 398
- mullins effect (node) 574
- multiaxial stress states 372
- multibody dynamics interface 704
- multiphysics
 - hygroscopic swelling 1120
 - magnetostriction 1109, 1123
 - piezoelectric devices 1105
 - piezoelectric effect 1121
- multiphysics coupling
 - fluid-membrane interaction, fixed geometry 1098
 - fluid-shell interaction 1087
 - fluid-solid interaction 1084, 1090
 - fluid-structure interaction 1093, 1124, 1128

- fluid-structure interaction, fixed geometry 1095
- fluid-structure interaction, pair 1126
- Joule heating and thermal expansion 1101
- thermal expansion (node) 1114
- thermal stress 1081
- multiphysics modeling 195
- MUMPS direct solvers 240
- Murnaghan material 336
- N**
 - Navarro-Herring creep 397
 - Navier-Stokes equations 494
 - nearly incompressible hyperelastic materials 323
 - nearly incompressible materials 106, 280, 296
 - Neo-Hookean material 326
 - Neumann boundary conditions
 - applied force 630
 - applied moment 632
 - no rotation (node)
 - beam interface 1002
 - shell and plate interfaces 814
 - nodes, common settings 30
 - nominal stress 282
 - nonassociated flow rule 368
 - nonlinear elastic material 514
 - nonlinear elastic material (node) 546
 - Norton equation 393
 - Norton-Bailey model 394
- O**
 - octahedral plane 278
 - Ogden material 330
 - one-way coupled model formulations 495
 - orientation, piezoelectric material 123
 - orthotropic materials
 - defining 292
 - elastic properties 543, 935
 - loss factor damping, and 616
 - Ottosen criterion 374
 - over-consolidation pressure 379
- P**
 - pair nodes
 - beam interface 978
 - membrane interface 931
 - solid mechanics 534
 - truss interface 1059
 - parametric analysis 24
 - penalized friction traction 500
 - penalty factor relaxation 178
 - penalty factors
 - contact node, and 713
 - contact pairs and 177
 - theory 500
 - penalty method 498
 - perfectly elastoplastic materials 361
 - perfectly plastic hardening 361
 - periodic boundary conditions 682, 684, 838
 - periodic condition (node)
 - shell 838
 - solid mechanics 682
 - periodic conditions, theory 488
 - phase (node) 726, 909
 - beam interface 1016
 - layered shell interface 909
 - shell and plate interfaces 852
 - solid mechanics 726
 - truss interface 1077
 - physics interfaces
 - connecting to DOF 74
 - physics interfaces, common settings 30
 - piezoelectric coupling 1105
 - piezoelectric crystal cut 123
 - piezoelectric devices multiphysics interface 1105
 - piezoelectric effect (node) 1121

- piezoelectric losses 131
- piezoelectric material (node) 565
- piezomagnetic coupling 569
- piezomagnetic coupling matrices 413
- pinned (node)
 - beam interface 1000
 - shell and plate interfaces 813
 - truss interface 1069
- pipe sections, beams 981
- plane stress and strain 66, 531
- planes, symmetry and constraints 93
- plastic deformation gradient 266
- plastic element 368, 370
- plastic flow rule 370
- plastic Green-Lagrange strain 266
- plastic multiplier 368
- plastic potential 368
- plasticity (node) 575
- plasticity models 575
- plate interface 779
 - theory 736
- plates
 - external loads 747
 - initial stresses and strains 796
 - linear elastic material 788
 - MITC shell formulation 742
 - prescribed acceleration 811
 - prescribed displacement/rotation 803
 - prescribed velocity 809
 - stress and strain calculations 747
 - symmetry and antisymmetry 745
 - thermal expansion 793
- point load (node) 667
 - beam interface 1008
 - layered shell interface 908
 - shell and plate interfaces 828
- point load on axis (node) 669
- point loads example 84
- point mass (node)
 - beam interface 1009
 - shell and plate interfaces 837
 - truss interface 1076
- point mass damping (node)
 - beam interface 1010
 - shell interface 838
 - truss interface 1077
- point nodes
 - beam interface 978
 - membrane interface 931
 - solid mechanics 534
 - truss interface 1059
- Poisson type equation 1032
- Poisson's ratio 107, 290, 296, 542, 672
- polynomial hyperelastic material 328
- power law 393
- Prager's hardening rule 364
- Prandtl stress function 1037
- predeformation (node) 678
 - layered shell interface 914
 - shell and plate interfaces 835
- prescribed acceleration (node) 644
 - beam interface 998
 - layered shell interface 894
 - shell and plate interfaces 811
- prescribed acceleration, interface (node)
 - 895
- prescribed displacement (node)
 - layered shell interface 889
 - solid mechanics 638
- prescribed displacement, interface
 - (node) 890
- prescribed displacement, theory 427
- prescribed displacement/rotation (node)
 - 627
 - beam interface 994
 - shell and plate interfaces 803

- theory 427
 - prescribed velocity (node) 642
 - beam interface 997
 - layered shell interface 892
 - shell and plate interfaces 809
 - solid mechanics 642
 - prescribed velocity, interface (node) 893
 - pressure loads 86
 - pressure-wave speeds 290, 543
 - prestressed analysis, eigenfrequency
 - study 167
 - primary creep 390
 - principal stresses 276, 278
 - principal stretches 319
 - principle of virtual work 446
 - Prony series 310
 - propagating modes 46
- Q**
- quaternion constraint 421
 - quaternion representation, of rigid connector 497
- R**
- radius of gyration 1027
 - rate independent plasticity 368
 - rate of strain tensor 262, 369
 - Rayleigh damping 149
 - rectangle sections, beams 981
 - reference coordinates 250
 - reference point for moment computation 99
 - refpnt variable 533
 - relaxation, of penalty factor 178
 - renaming
 - displacement DOF 75
 - representative volume element 215, 496
 - resonant frequency 149
 - results evaluation in a laminate 776
 - results evaluation, for shells 751
 - results evaluation, layered shells 867
 - right Cauchy-Green deformation tensor
 - 257, 319
 - right polar decomposition 256
 - right stretch tensor 256
 - rigid body 419
 - rigid connection type, shells 754
 - rigid connector 497
 - rigid connector (node) 689, 1013
 - layered shell interface 916
 - shell and plate interfaces 846
 - rigid connector theory 427
 - rigid connector, interface (node) 917
 - rigid connectors
 - kinematic constraints and 97
 - moments and 85
 - rigid domain (node) 622
 - layered shell interface 887
 - theory 419
 - rigid domains
 - angular acceleration 423
 - angular velocity 422
 - dynamics 423
 - rigid motion suppression (node) 657
 - layered shell interface 902
 - ring load (node) 668
 - shell and plate interface 830
 - rock mass 377
 - rock types 376
 - rocks 375
 - rocks (node) 593
 - roller (node) 650
 - layered shell interface 898
 - roller, interface (node) 899
 - rotated coordinate system 126
 - rotating frame (node) 662
 - layered shell interface 906
 - rotation matrix 421
 - rotation, rigid body 420
 - rotational degrees of freedom 943

- rotational joints, beams 98
- S**
 - safety (node) 436, 608, 800, 1045
 - layered shell interface 886
 - Saint-Venant torsion 1041
 - saturated cohesive soils 345
 - scaling of eigenvectors 43
 - second Piola-Kirchhoff stress 160, 252, 274
 - secondary creep 390
 - section orientation (node) 985
 - selecting
 - solvers 239
 - serendipity elements 71
 - shape factors 375
 - shape memory alloy 403
 - shear area 1029
 - shear correction factor computation 774
 - shear modulus expression 290, 542, 672
 - shear stresses 276, 349
 - shear-wave speeds 290, 543
 - shell
 - coupling to a beam 78
 - coupling to a solid 76
 - shell connection (node)
 - beams 1012
 - solid mechanics 709
 - shell interface 779
 - theory 736
 - shells
 - external loads 747
 - initial stresses and strains 796
 - linear elastic material 788
 - MITC shell formulation 742
 - prescribed acceleration 811
 - prescribed displacement/rotation 803
 - prescribed velocity 809
 - stress and strain calculations 747
 - symmetry and antisymmetry 745
 - thermal expansion 793
 - shift function 571
 - sign conventions 275
 - simply supported (node) 816
 - singular loads 84
 - size factors 375
 - skew-symmetric part 262
 - slit boundary 217
 - SLS model 312
 - small plastic strain 367
 - soil deformation theory 380
 - soil plasticity (node) 590
 - solid
 - coupling to a beam 76–77
 - coupling to a shell 76
 - solid connection (node) 842
 - solid mechanics
 - damping 614
 - edge loads 666
 - initial stresses and strains 598
 - prescribed acceleration 644
 - prescribed velocity 642
 - solid mechanics interface 530
 - theory 248
 - solid-shell connection (node) 1128, 1130, 1134
 - solver methods
 - augmented Lagrangian 498
 - penalty method 498
 - solver parameters 239
 - solver settings 239
 - SOR line solvers 241
 - sources and destinations 174
 - spatial coordinates 250
 - spatial stress tensor 160
 - spatially mobilized planes (STP) 355
 - spin tensor 262
 - spring constant 193

- spring foundation 635
- spring foundation (node) 635, 670, 702
 - layered shell interface 910–911
 - shell and plate interfaces 831
- spring foundation, solid mechanics 193
- spring foundation, theory 479
- spring-damper (node) 1068
- St Venant torsion 1039
- St Venant-Kirchhoff material 326
- St. Venant's principle 84
- standard linear solid model 312
- standard settings 30
- static frictional coefficients 503
- stationary solvers 239
- steady-state creep 390
- steady-state stiffness 310
- storage modulus 153, 317
- Storakers material 331
- stored energy 514
- straight edge constraint (node) 1062
- straight edges 1053
- strain energy density 338, 514
- strain-displacement, trusses 1050
- strain-displacement/rotation 951
- strains, membranes 925
- stress
 - Cachy 274
 - first Piola-Kirchhoff 274
 - second Piola-Kirchhoff 274
- stress and strain, piezoelectric devices
 - 145, 407, 413
- stress components, beam cross sections 1031
- stress evaluation, beams 956
- stress linearization (node) 729
- stress stiffening 924
- stresses, membranes 925
- stress-strain relation
 - beams 951
 - trusses 1051
- study steps, geometric nonlinearity and 163
- study types
 - eigenfrequency 449
 - frequency domain, solid mechanics 448
 - parametric 24
 - surface traction and reaction forces 101
- symmetric matrices 239
- symmetry (node)
 - beam interface 1003
 - layered shell interface 900
 - shell and plate interfaces 817
 - solid mechanics 652
 - truss interface 1072
- symmetry constraints 92
- symmetry plane (node) 820
 - solid mechanics 654

T

- tangent modulus 362
- tangential strains 927
- technical support, COMSOL 34
- temperature loads 86
- tensile meridian 277
- tensile meridians 350, 352, 357, 373–374
- tension cut-off 357
- tertiary creep 390
- theory
 - beam cross sections 1026
 - beam interface 942
 - fluid-structure interaction 494
 - layered shell interface 856
 - membrane interface 924
 - shell and plate interfaces 736
 - solid mechanics interface 248
- theory background of laminated shell 762
- thermal expansion
 - loads and 86

- thermal expansion (for constraints)
 - (node)
 - layered shell interface 898
- thermal expansion (multiphysics coupling) 1114
- thermal expansion (node) 594
 - beam interface 990
 - layered shell interface 881
 - shell and plate interfaces 793
 - solid mechanics 594, 648, 807, 1001, 1070
- thermal expansion, hyperelastic materials and 320
- thermal expansion, Joule heating and 1101
- thermal strain
 - beams 952
- thermal stress interface 1081
 - modeling 195
- thermal-electric-structural interaction 196
- thermoelastic materials, defining 295
- thermorheologically simple viscoelastic materials 315
- thin elastic layer (node) 675
 - layered shell interface 912
- thin elastic layer, interface (node) 913
- thin elastic layer, solid mechanics 193
- thin elastic layer, theory 481
- thin-film damping (node) 688
- thread boundary selection (node) 725
- three-parameter model 312
- torional shear stress 1041
- torsion, beam cross sections 1037
- torsional constants and moments 955
- total Lagrangian formulation 24
- total loads 87
- t-profile sections, beams 981
- tractions 160
- transient creep 390
- translation, rigid body 420
- Tresca effective stress variable 344
- Tresca stress 343
- Tresca yield criterion 343
- triaxial conditions 373
- triaxial data 375
- TRS material 315
- true stress tensor 160
- truss interface 1058
- trusses
 - harmonic perturbation 728, 853, 1017
 - initial stresses and strains 1066
 - linear elastic material 1063
 - straight edge 1053
 - strain-displacement 1050
- two-point tensor 255

U

- undrained shear strength 344
- uniaxial compression 372, 374, 376, 444
- uniaxial tension 343, 593
- units, loads and 81
- u-profile sections, beams 981
- user defined material 336
- using
 - coordinate systems 251
 - predefined variables 99
 - spatial and material coordinates 250
 - weak constraints 100
- uspring variable 193

V

- Varga material 332
- variables
 - cross section data, beams 1022
 - deformation gradient tensor 256
 - density 256
 - effective creep strain 393
 - effective plastic strain 342, 364
 - elastic Green-Lagrange tensor 265

- elastic right Cauchy-Green tensor 265
- elastic volumetric strain 381
- elastic, inelastic, and total volume ratio
 - 265
- elcontact 499
- Green-Lagrange tensor 258
- invariants 336
- isochoric elastic Green-Lagrange strain tensor 323
- isochoric-elastic Cauchy-Green deformation tensor 322
- isochoric-elastic right Cauchy-Green deformation tensor 323
- material and spatial coordinates 251
- predefined 99
- principal elastic stretches 337
- refpnt 533
- right Cauchy-Green deformation tensor 258
- right stretch tensor 256
- rotation tensor 256
- stresses, beams 1024
- thermal stretch and thermal volume ratio 321
- Tresca effective stress 344
- volumetric plastic strain 342, 379
- vdamper variable 194
- Vicat softening temperature 316
- Villari effect 412
- viscoelastic materials
 - definition 106
 - frequency domain analysis and damping 317
 - temperature effects 315
 - theory 308
- viscoelasticity (node) 570
 - layered shell interface 878
- viscoplasticity (node) 583, 588
 - layered shell interface 880
- viscous damping 153, 194
- Voigt form 124
- Voigt notation 291, 616
- volume reference density 199
- volume-preserving contributions 321
- volumetric contributions 321
- volumetric creep strain rate 392
- volumetric plastic strain rate 342
- volumetric plastic strain variable 379
- volumetric strain energy 337
- volumetric strain energy density 323
- von Mises criterion 342–343
- von Mises effective stress 1042
- von Mises stress 342

W

- warping function 1037
- warping, beam cross sections 1039
- wave speeds 543, 935, 989
- waveguide 46
- weak constraints, using 100
- websites, COMSOL 35
- Weertman creep 398
- Willam-Warnke criterion 373
- WLF shift functions 315, 571

Y

- yield function 342
- yield functions 350
- yield stress levels 576, 584
- yield surface 368
- yield surfaces 340, 351
- Young's modulus expression 290, 542, 672

Z

- Zener model 312