Structural Mechanics Module

User’s Guide

5.3

COMSOL

Structural Mechanics Module User's Guide

Version: COMSOL 5.3

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

Contents

Chapter 1: Introduction

Chapter 2: Structural Mechanics Modeling

Introduction to Material Models **90**

Modeling Piezoelectric Problems **96**

Modeling Magnetostrictive Materials **116**

Mechanical Damping and Losses **119**

Modeling Geometric Nonlinearity **127**

Contact Modeling **141**

Springs and Dampers **160**

Defining Multiphysics Models **162**

Thermally Coupled Problems **165**

Chapter 3: Structural Mechanics Theory

Chapter 4: Solid Mechanics

Chapter 5: Shell and Plate

Chapter 6: Membrane

Chapter 7: Beam

Chapter 8: Beam Cross Section

Chapter 9: Truss

Chapter 10: Multiphysics Interfaces and Couplings

Chapter 11: Glossary

1

Introduction

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

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

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

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

In the *COMSOL Multiphysics Reference Manual:*

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

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Fluid Flow				
Fluid-Structure Interaction		fsi	3D, 2D, 2D axisymmetric	stationary; stationary, one-way coupled; time dependent; time dependent, one-way coupled
Fluid-Structure Interaction, Fixed Geometry[2]		—	3D, 2D, 2D axisymmetric	stationary; time-dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Structural Mechanics				
Solid Mechanics[1]		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; prestressed analysis, eigenfrequency; mode analysis; time dependent; time-dependent modal; frequency domain; frequency-domain modal; prestressed analysis, frequency domain; modal reduced order model; 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
Shell		shell	3D	stationary; eigenfrequency; prestressed analysis, eigenfrequency; time dependent; time-dependent modal; frequency domain; frequency-domain modal; prestressed analysis, frequency domain; modal reduced order model; linear buckling
Plate		plate	2D	stationary; eigenfrequency; prestressed analysis, eigenfrequency; time dependent; time-dependent modal; frequency domain; frequency-domain modal; prestressed analysis, frequency domain; modal reduced order model; linear buckling

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
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
Truss		truss	3D, 2D	stationary; eigenfrequency; prestressed analysis, eigenfrequency; time dependent; time-dependent modal; frequency domain; frequency-domain modal; prestressed analysis, frequency domain; modal reduced order model; linear buckling
Membrane		mbrn	3D, 2D axisymmetric	stationary; eigenfrequency; prestressed analysis, eigenfrequency; time dependent; time-dependent modal; frequency domain; frequency-domain modal; prestressed analysis, frequency domain
Piezoelectric Devices[2]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; time-dependent modal; frequency domain; frequency-domain modal; small-signal analysis, frequency domain; prestressed analysis, eigenfrequency; prestressed analysis, frequency domain; modal reduced order model

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Magnetostriction[2]		—	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent; frequency domain; small-signal analysis, frequency domain; prestressed analysis, eigenfrequency; prestressed analysis, frequency domain

[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			GEOMETRY LEVEL			
	NAME	DEPENDENT VARIABLES	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	$\mathbf{u}_{\mathrm{solid}}$, $\mathbf{u}_{\mathrm{fluid}}$, p	√	√	√	√
Fluid-Structure Interaction, Fixed Geometry	n/a*	$\mathbf{u}_{\mathrm{solid}}$, $\mathbf{u}_{\mathrm{fluid}}$, 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 (as long as you have a license), 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 many of the features on the GUI. To learn more about a node in the Model Builder, or a window on the Desktop, click to highlight a node or window, then press F1 to open the Help window, which then displays information about that feature (or click a node in the Model Builder followed by the **Help** button (?). This is called *topic-based* (or *context) help*.

Win

To open the **Help** window:

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

Mac Linux

To open the **Help** window:

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

Opening the Documentation Window

Win

To open the **Documentation** window:

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

Mac Linux

To open the **Documentation** window:

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

THE APPLICATION LIBRARIES WINDOW

Each application includes documentation with the theoretical background and step-by-step instructions to create a model application. The applications are available in COMSOL as MPH-files that you can open for further investigation. You can use the step-by-step instructions and the actual applications as a template for your own modeling and applications. In most models, SI units are used to describe the relevant properties, parameters, and dimensions in most examples, 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 application and its properties, including options to open it or a PDF document.

🔍

The Application Libraries Window in the *COMSOL Multiphysics Reference Manual*.

Opening the Application Libraries Window

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

Win

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

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

Mac

Linux

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

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

CONTACTING COMSOL BY EMAIL

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

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.

COMSOL ONLINE RESOURCES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
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, the you will find a discussion 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-Structure Interaction Interface, found under the Fluid Flow branch, is described in this chapter. The physics interface combines fluid flow with solid mechanics to capture the interaction between the fluid and the solid structure.
- The Fluid-Structure Interaction, Fixed Geometry Interface, found under the Fluid Flow branch, is described in this chapter. The interface can be used to model phenomena where a fluid and a deformable solid structure affect each other.

2

Structural Mechanics Modeling

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

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

In this chapter:

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 Pre-Tension 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 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.

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. If output of *mass participation factors* is required, then **Scaling of eigenvectors** must be set to **Mass matrix**. This means that the eigenmodes **U** are orthogonalized with respect to the mass matrix M so that

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

This is a common choice for the scaling of eigenvectors within the structural mechanics field.

The mass (or 'modal') participation factor for mode i in direction j, r_{ij}, is defined as

$$r_{ij} = \mathbf{U}_i^T M \mathbf{d}_j$$

Here, $\mathbf{d}_j$ is a vector containing unity displacement in all degrees of freedom representing translation in direction j. The mass participation factor gives an indication of to which extent a certain mode might respond to an excitation in that direction.

The mass participation factors have the important property that when their squares for a certain direction are summed over all modes, this sum approaches the total mass of the model:

$$\sum_{i=1}^{n} r_{ij}^2 = m_{\mathrm{tot}}$$

If you use a modal superposition method to solve a forced response problem, then in practice you do not solve for all possible modes but just a limited number. This property if the mass participation factors can then be used for investigating how well a certain number of selected modes represent the total mass of the system.

The mass participation factors are available as a global variables, and these can be shown in a table using a **Global Evaluation** node under **Derived Values** in the **Results** branch, for example. The participation factor variables are available as predefined variables in the **Solver** submenu.

For an example showing how to compute 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 and Tables

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 2-1: 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)

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,

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

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*. With the Prestressed Analysis, Eigenfrequency study type you can compute eigenfrequencies taking this effect into account.

	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[N]*(1+0.5*i)/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 `real(u)`. This gives the displacement at zero phase angle.
- Plot `imag(u)`. This gives the displacement at a phase angle of 90 degrees.
- Plot `abs(u)`. This gives the amplitude of the displacement.
- Plot `arg(u)`. This gives the phase angle of the displacement.

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

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
- `colid.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. The Prestressed Analysis, Frequency Domain study type is designed to take such effects into account.

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 two predefined study types for modal superposition: Time-Dependent Modal and Frequency-Domain Modal.

Both these study types consist of two study steps: One step for computing the eigenfrequencies and one step for the modal response.

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. This means that there are no restrictions on the type of damping that can be used in a modal superposition analysis, as it would have been the case if the modal equations were assumed to be totally decoupled.

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 `linper()` 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 2-2.

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

STUDY TYPE	STUDY STEP	PERTURBATION
Frequency Domain	Frequency Domain	No

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

STUDY TYPE	STUDY STEP	PERTURBATION
Prestressed Analysis, Frequency Domain	Stationary Frequency-Domain, Perturbation	No Yes
Frequency-Domain, Modal	Eigenfrequency Frequency-Domain, Modal	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

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

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

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

$$K\hat{\mathbf{u}}_i = \omega_i^2 M\hat{\mathbf{u}}_i \tag{2-3}$$

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

$$\hat{\mathbf{u}}_j^T M\hat{\mathbf{u}}_i = 0 \qquad i \neq j,\ \omega_i \neq \omega_j \tag{2-4}$$

$$\hat{\mathbf{u}}_j^T K\hat{\mathbf{u}}_i = 0 \qquad i \neq j,\ \omega_i \neq \omega_j \tag{2-5}$$

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

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

Then consider the following matrix:

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

From Equation 2-4 it is clear that this is a diagonal matrix. Similarly from Equation 2-5 it is clear that U^TKU is also diagonal.

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

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

This equation can also be expressed in the form:

$$\mathbf{u} = U\mathbf{a} \qquad (2\text{-}6)$$

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:

$$U^TMU\ddot{\mathbf{a}} + U^TDU\dot{\mathbf{a}} + U^TKU\mathbf{a} = U^T\mathbf{F} \qquad (2\text{-}7)$$

It has already been established that the matrices U^TMU and U^TKU 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 $U^TMU = I$. 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

$$U^TKU = \mathrm{diag}(\omega_i^2)$$

where diag(ω_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

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

Thus if *mass matrix* scaling is used Equation 2-7 takes the form

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

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

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

Here $m_{\mathrm{eff},i}$ is the effective mass of the i:th mode, $c_{\mathrm{eff},i} = 2m_{\mathrm{eff},i}\xi_i\omega_i$ is the effective damping parameter for the mode, and $k_{\mathrm{eff},i}$ 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 $i = 1, \ldots, m$ 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

$$U'^T MU'\ddot{\mathbf{a}}' + U'^T DU'\dot{\mathbf{a}}' + U'^T KU'\mathbf{a}' = U'^T\mathbf{F} \tag{2-10}$$

The matrices $U'^T MU'$, $U'^T DU'$, and $U'^T KU'$ 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

$$U'_l{}^T MU'_r\ddot{\mathbf{a}}' + U'_l{}^T DU'_r\dot{\mathbf{a}}' + U'_l{}^T KU'_r\mathbf{a}' = U'_l{}^T\mathbf{F}$$

where U'_l is the n-by-m matrix containing the left eigenvectors chosen for the modal analysis.

The matrices $U'_l{}^T MU'_r$, $U'_l{}^T DU'_r$, and $U'_l{}^T KU'_r$ 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 1** 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 $U'_l{}^T MU'_r$ the stiffness matrix corresponds to $U'_l{}^T KU'_r$, and the damping matrix corresponds to $U'_l{}^T DU'_r$. 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 1>Solver Configurations>Solver 1>Eigenvalue Solver 1** 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 $\mathbf{f}$,

$$K\mathbf{u} = (K_{\mathrm{L}} + K_{\mathrm{NL}})\mathbf{u} = \mathbf{f}$$

Here the total stiffness matrix, K, has been split into a linear part, K_{L}, and a nonlinear contribution, K_{NL}.

In a first order approximation, K_{NL} is proportional to the stress in the structure and thus to the external load. So if the linear problem is solved first for an arbitrary initial load level $\mathbf{f}_0$,

$$K_{\mathrm{L}}\mathbf{u}_0 = \mathbf{f}_0$$

then the nonlinear problem can be approximated as

$$(K_{\mathrm{L}} + \lambda K_{\mathrm{NL}}(\mathbf{u}_0))\mathbf{u} = \lambda \mathbf{f}_0$$

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.

$$(K_{\mathrm{L}} + \lambda K_{\mathrm{NL}}(\mathbf{u}_0))\mathbf{u} = 0$$

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

The 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.

!	You should 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 pre-load study step.

📋	Be aware that for some structures, the true buckling load can be significantly smaller that what is computed using a linearized analysis. This phenomenon is sometimes called *imperfection sensitivity.* Small deviations from the theoretical geometrical shape can then have a large impact on the actual buckling load. This is especially important for curved shells.

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

Studies and Solvers and Linear Buckling in the *COMSOL Multiphysics Reference Manual*

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

Bolt Pre-Tension Study

The **Bolt Pre-Tension** 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 Pre-Tension** study step as the first step in a study in order to simulate the state after the assembly of a bolted joint. You can the add any other types of study steps for computing the effects of the service loads.

Pre-tensioned Bolts

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

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.

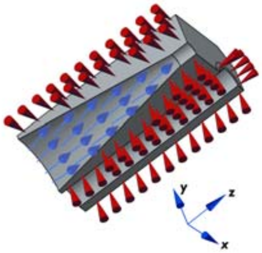


Figure 2-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.

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.

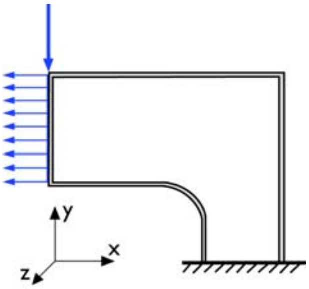


Figure 2-2: Plane stress models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

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.

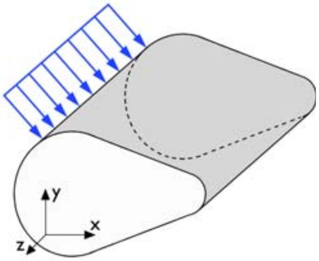


Figure 2-3: A geometry suitable for plane strain analysis.

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 (`phi`), 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 $\varphi = 0$, $r \geq 0$. 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.

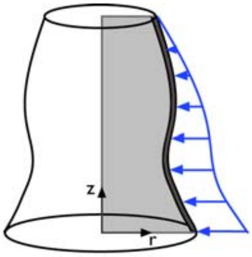


Figure 2-4: Rotating a 2D geometry to recover a 3D solid.

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 is space.

In most applications, a membrane must be pre-tensioned 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 trade-off between model size and accuracy.

It is well known that using first-order shape functions in solid mechanics will give an overly stiff solution, unless a very fine mesh is used. This is especially noticeable for triangular and tetrahedral elements.

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 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 differs 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
- The Nodal Serendipity Element

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:

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

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 **Nearly incompressible material** check box 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 changes in material properties.

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, 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 Edge to Solid Boundary (2D)
- 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.

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 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.

Beam Edge to Solid Boundary (2D)

A beam can also be coupled to a solid by adding a Solid-Beam Connection multiphysics coupling with **Connection type** set to **Solid and beam parallel boundaries**. This connection is used for adding a beam on top of a solid as a ‘cladding’. The boundaries are assumed to be parallel.

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 are 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.

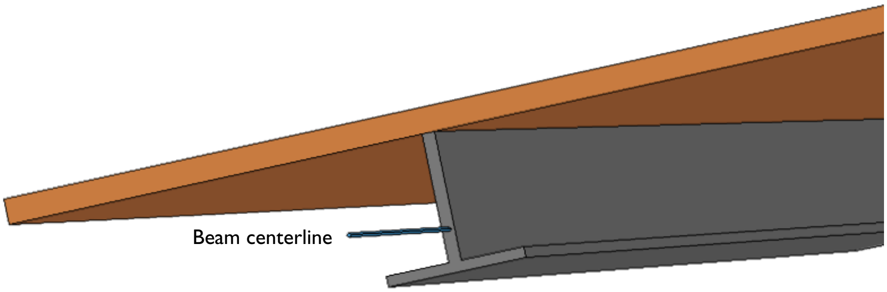
- Examples of all types of couplings between shells and beams are shown in *Connecting Shells and Beams*: Application Library path **Structural_Mechanics_Module/Tutorials/shell_beam_connection**
- An example of couplings between shells and solids is shown in *Connecting Shells and Solids*: Application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

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

or equivalently as

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

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

To create the coupling:

1 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[lbf/in^2]`.

PREDEFINED AND CUSTOM COORDINATE SYSTEMS

In this module, different predefined coordinate systems are available when loads 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 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.

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.

In the *COMSOL Multiphysics Reference Manual*:

- Physics Symbols
- Using Units
- Coordinate Systems

For an example showing how to examine the load distribution, see *Bracket—Static Analysis*: Application Library path **Structural_Mechanics_Module/Tutorials/bracket_static**. This is also the first example used in the *Introduction to the Structural Mechanics Module*.

Load Cases

For a Stationary study, you can define load cases and constraint cases. A load or constraint can be assigned to a load or constraint group, and then used conditionally.

For most load types, the load case acts as a simple multiplier, but some cases need a more detailed discussion:

- A **Prescribed Displacement** or **Prescribed Displacement/Rotation** node can be assigned both a constraint group and a load group. You can use the constraint group to switch off the whole constraint. The load group acts as a multiplier to non-zero 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 non-zero 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 a 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.

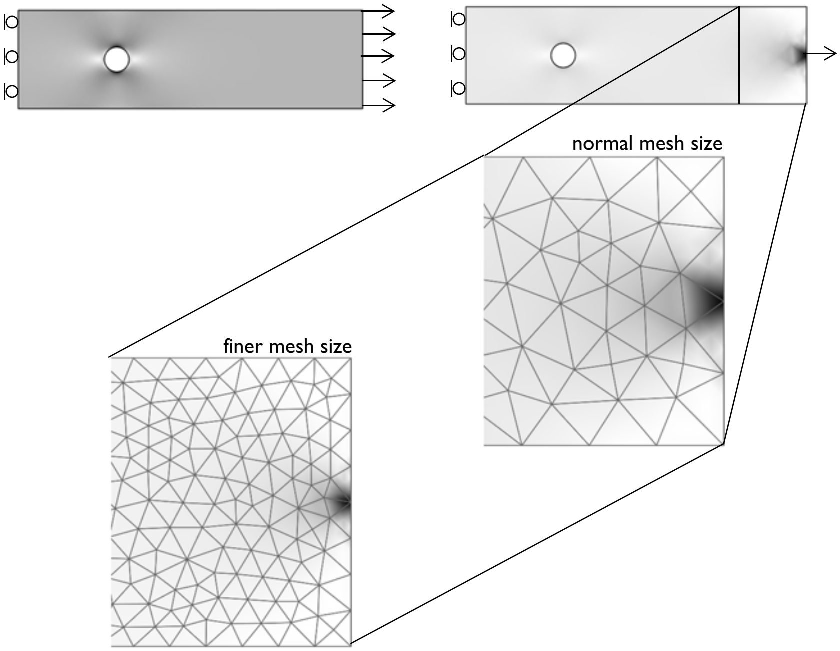


Figure 2-5: A plate with a hole subject to a distributed load (left) and a point load (right).

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.

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

$$K_r = \rho\omega^2 r \tag{2-11}$$

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-Structural 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
- 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 2-3: 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 round-off 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 non-zero 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:

$$u_\mathrm{p} = \frac{v_\mathrm{p}}{i\omega}$$

$$u_\mathrm{p} = \frac{-a_\mathrm{p}}{\omega^2}$$

where ω is the angular frequency.

In the case of a time-dependent analysis, the prescribed displacement is obtained as

$$u_\mathrm{p}(t) = u_0(t_0) + \int_{t_0}^{t} v_\mathrm{p}(\tau)d\tau$$

or

$$u_\mathrm{p}(t) = u_0(t_0) + \int_{t_0}^{t}\left(v_0(t_0) + \int_{t_0}^{\tau} a_\mathrm{p}(\zeta)d\zeta\right)d\tau$$

where u_0 and v_0 are is given by the initial conditions. It is not possible to set explicit initial conditions, but if initial values are taken from a previous study, they will be respected. In order to compute the integrals, u_p is introduced as a separate degree of freedom which is solved for by adding an extra ODE.

As prescribing the velocity or acceleration in time domain comes with an extra cost, you should always consider using a prescribed displacement instead. As long as the time history of the velocity or acceleration is a known a priori and does not depend on the solution itself, this is possible.

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

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.

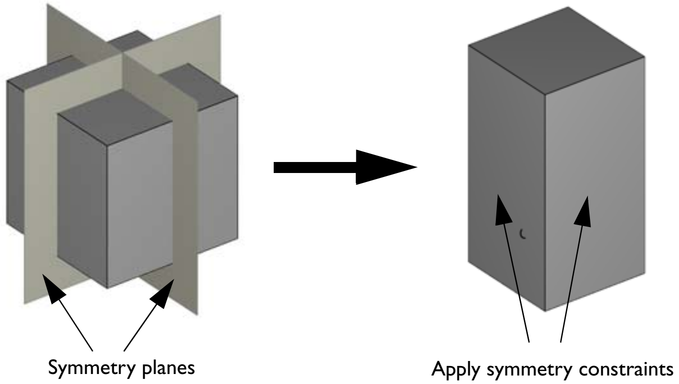


Figure 2-6: If the geometry exhibits two symmetry planes, model a quarter of the geometry by using the Symmetry feature for the two selected surfaces.

!	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 non-symmetric 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.

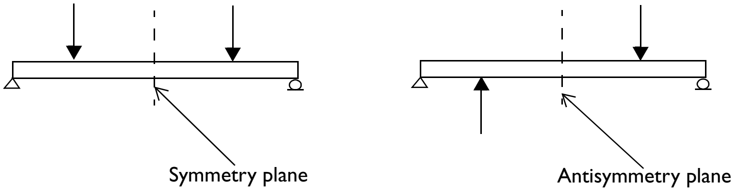


Figure 2-7: Symmetry plane (left) and antisymmetry plane (right).

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 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*.

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.

For an example showing how to force a boundary to remain plane, but still allow it to translate in its normal direction, see *Thermo-Mechanical Analysis of a Surface-Mounted Resistor*: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/ surface_resistor**.

In the 3D and 2D Solid Mechanics interfaces and in the Shell interface there is a special constraint called a Rigid Connector. A rigid connector is applied to one or more boundaries or edges 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**. 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.
- 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 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_lm`, where *x* is the name of the constrained degree of freedom (as, for example, `u_lm` and `v_lm`). The extension `lm` stands for *Lagrange multipliers*. It is only possible to evaluate reaction forces on constrained boundaries in the directions of the constraints.

To compute the total reaction force on a boundary, integrate one of the variables `x_lm` using **Volume Integration**, **Surface Integration**, or **Line Integration** under **Derived Values**.

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 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*.Trac*x*, 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 check box **Compute boundary fluxes** in the **Discretization** section is selected for the Solid Mechanics interface.

In case of geometric nonlinearity, the two types of traction variables are interpreted differently. The *interface*.Ta*x* variables are based on Cauchy stress, and contains a force per current area. If you integrate them you must use the spatial frame. The *interface*.Trac*x* variables are based on 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*.iTracu*x* and *interface*.iTracd*x*. 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*.iTracu*x* and*interface*.iTracd*x* 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
- About the Material Databases for the Structural Mechanics Module

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 materials modeling, which is described in detail in the section Modeling Piezoelectric Problems.

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.

In the *COMSOL Multiphysics Reference Manual*:

- Working with External Materials
- External Material

In Table 2-4 you can find an overview of the families of materials, and their applicability in the various structural mechanics interfaces.

TABLE 2-4: 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	-	-
Piezoelectric Material	X	-	-	-	-
Magnetostrictive Material	X	-	-	-	-
Viscoelasticity	X	-	X	-	-
Plasticity	X	-	X	-	X
Soil Plasticity	X	-	-	-	-
Creep	X	-	X	-	-
Concrete	X	-	-	-	-
Rocks	X	-	-	-	-
Cam-Clay Material	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 **Nearly incompressible material** check box is selected in the **Settings** window for the material, 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 interpolated pressure 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 **Hyperelastic Material** and **Plasticity** nodes are available with the Nonlinear Structural Materials Module. The **Plasticity** node is also available with the Geomechanics Module.

The order of the shape function for the auxiliary pressure variable should be one order less than that for the displacements. Thus, it is not recommended to use linear elements for the displacement variables in the domains, where the mixed formulation is turned on. Also note that some iterative solvers do not work well together with *mixed formulation* because the stiffness matrix becomes indefinite.

For an isotropic linear elastic material, the second Piola-Kirchhoff 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

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

The auxiliary dependent variable p_w is set equal to p using the equation

$$\frac{p_w}{\kappa} - \frac{p}{\kappa} = 0 \tag{2-12}$$

where κ is the bulk modulus.

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

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, and the pressure p with:

$$p_p = -\frac{1}{3}\mathrm{trace}(S)$$

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**.

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:

$$\begin{aligned} S &= s_E T + d^T \mathbf{E} \\ \mathbf{D} &= dT + \varepsilon_T \mathbf{E} \end{aligned} \tag{2-13}$$

Here, the naming convention used in piezoelectricity theory is assumed: S is the strain, T is the stress, $\mathbf{E}$ is the electric field, and $\mathbf{D}$ is the electric displacement field. The material parameters s_E, d, and ε_T, correspond to the material compliance, the coupling properties and the permittivity. These quantities are tensors of rank 4, 3, and 2 respectively. The tensors however are highly symmetric for physical reasons, and they can be represented as matrices within an abbreviated subscript notation, which is usually more convenient. In the 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 **T)**. This makes the names consistent with those used in the other structural mechanics interfaces.

Equation 2-13 will, using the notation from structural mechanics, then read

$$\begin{aligned} \varepsilon &= s_E\sigma + d^T\mathbf{E} \\ \mathbf{D} &= d\sigma + \varepsilon_0\varepsilon_{\mathrm{r}T}\mathbf{E} \end{aligned} \tag{2-14}$$

Equation 2-13 (or Equation 2-14) 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:

$$\begin{aligned} \sigma &= c_E\varepsilon - e^T\mathbf{E} \\ \mathbf{D} &= e\varepsilon + \varepsilon_0\varepsilon_{\mathrm{r}S}\mathbf{E} \end{aligned} \tag{2-15}$$

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 2-13 or Equation 2-15, 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 2-5). In the absence of a user-defined coordinate system, the local system corresponds to the global X, Y, and Z coordinate axes. When an alternative coordinate system is selected this system defines the orientation of the crystal axes. This is the mechanism used in COMSOL to define a particular crystal cut, and typically it is necessary to calculate the appropriate Euler angles for the cut (given the thickness orientation for the wafer). All piezoelectric material properties are defined using the Voigt form of the abbreviated subscript notation, which is universally employed in the literature (this differs from the standard notation used for the Solid Mechanics interface material properties). The material properties are defined in the material frame, so that if the solid rotates during deformation the material properties rotate with the solid. See Modeling Geometric Nonlinearity.

Crystal cuts are usually defined by a mechanism introduced by the IEEE/IRE standards. Both standards use a notation that defines the orientation of a virtual slice (the plate) through the crystal. The crystal axes are denoted X, Y, and Z and the plate, which is usually rectangular, is defined as having sides l, w, and t (length, width, and thickness). Initially the plate is aligned with respect to the crystal axes and then up to three rotations are defined, using a right-handed convention about axes embedded along the l, w, and t sides of the plate. Taking AT cut quartz as an example, the IEEE 1978 standard defines the cut as: (YXl) −35.25°. The first two letters in the bracketed expression always refer to the initial orientation of the thickness and the length of the plate. Subsequent bracketed letters then define up to three rotational axes, which move with the plate as it is rotated. Angles of rotation about these axes are specified after the bracketed expression in the order of the letters, using a right-handed convention. For

AT cut quartz only one rotation, about the l axis, is required. This is illustrated in Figure 2-9. Note that within the 1949 IRE Standard AT cut quartz is denoted as: (YXl) +35.25°, since the X-axis is rotated by 180° in this convention and positive angles therefore correspond to the opposite direction of rotation (see Figure 2-8). Table 2-6 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 2-8 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 2-5 summarizes the signs for the important matrix elements under the two conventions. Table 2-6 shows the different definitions of the crystal cuts under the two conventions.

TABLE 2-5: SIGNS FOR THE MATERIAL PROPERTIES OF QUARTZ, WITHIN THE TWO STANDARDS COMMONLY EMPLOYED

	IRE 1949 STANDARD		IEEE 1978 STANDARD	
MATERIAL PROPERTY	RIGHT HANDED QUARTZ	LEFT HANDED QUARTZ	RIGHT HANDED QUARTZ	LEFT HANDED QUARTZ
s_{14}	+	+	-	-
c_{14}	-	-	+	+
d_{11}	-	+	+	-
d_{14}	-	+	-	+
e_{11}	-	+	+	-
e_{14}	+	-	+	-

TABLE 2-6: CRYSTAL CUT DEFINITIONS FOR QUARTZ CUTS WITHIN THE TWO STANDARDS COMMONLY EMPLOYED AND THE CORRESPONDING EULER ANGLES FOR DIFFERENT ORIENTATIONS OF THE CRYSTAL THICKNESS

STANDARD	REPRESENTATION	AT CUT	BT CUT
IRE 1949	Standard	(*YXl*) +35.25°	(*YXl*) *–49°*
	Y-thickness Euler angles	(*ZXZ*: 0°,–35.25°,0°)	(*ZXZ*: 0°,49°,0°)
	Z-thickness Euler angles	(*ZXZ*: 0°,–125.25°,0°)	(*ZXZ*: 0°, –41°,0°)
IEEE 1978	Standard	(*YXl*) –35.25°	(*YXl*) +49°
	Y-thickness Euler angles	(*ZXZ*: 0°, 35.25°,0°)	(*ZXZ*: 0°, –49°,0°)
	Z-thickness Euler angles	(*ZXZ*: 0°, –54.75°,0°)	(*ZXZ*: 0°, –139°,0°)

When defining the material orientation, it is necessary to consider the orientation of the plate with respect to the global coordinate system in addition to the orientation of the plate with respect to the crystallographic axes. Consider the example of AT cut quartz in Figure 2-9. 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 2-10 shows how to define the local coordinate system in this case. Figure 2-11 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.

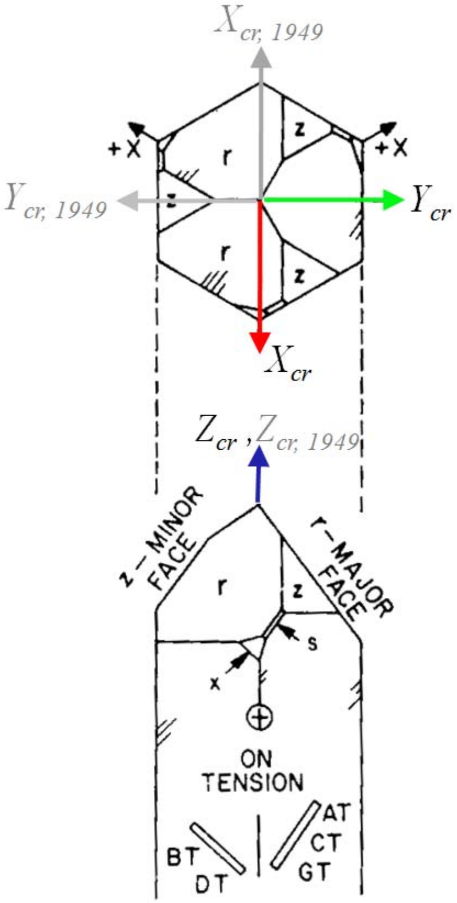


Figure 2-8: Crystallographic axes defined for right-handed quartz in COMSOL and the 1978 IEEE standard (color). The 1949 standard axes are shown for comparison (gray).

Figure 2-8 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.

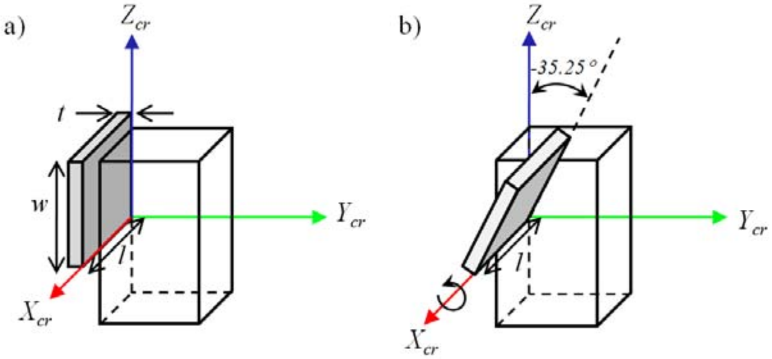


Figure 2-9: Definition of the AT cut of quartz within the IEEE 1978 standard. The AT cut is defined as: (YXl) −35.25°. The first two bracketed letters specify the initial orientation of the plate, with the thickness direction, t, along the crystal Y axis and the length direction, l, along the X axis. Then up to three rotations about axes that move with the plate are specified by the corresponding bracketed letters and the subsequent angles. In this case only one rotation is required about the l axis, of −35.25° (in a right-handed sense).

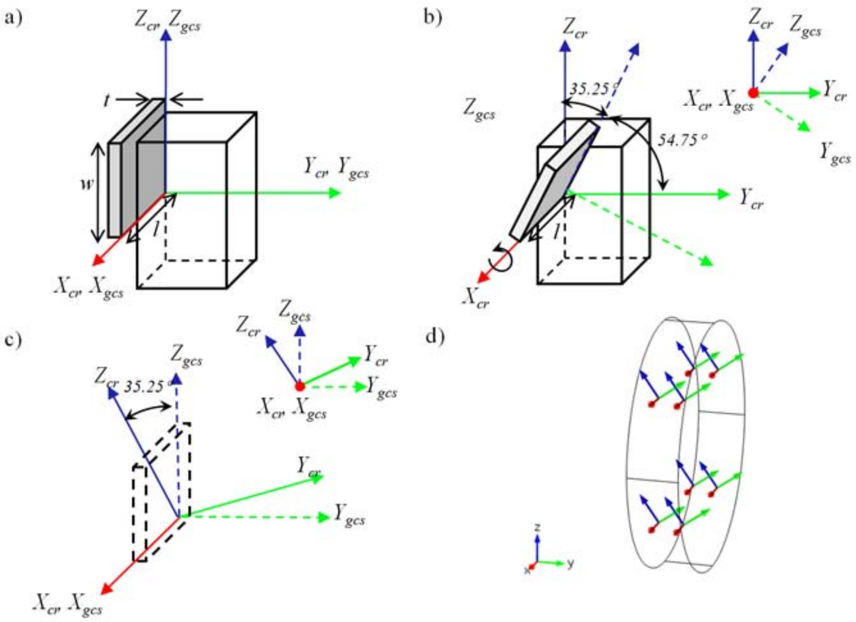


Figure 2-10: Defining an AT cut crystal plate within COMSOL, with normal in the global Y-direction. Within the 1978 IEEE standard the AT cut is defined as (YXl) -35.25°. Start with the plate normal or thickness in the Y_{cr} *direction (a) and rotate the plate –35.25° about the l axis (b). The global 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°, 0).(d) shows a coordinate system for this system in COMSOL.*

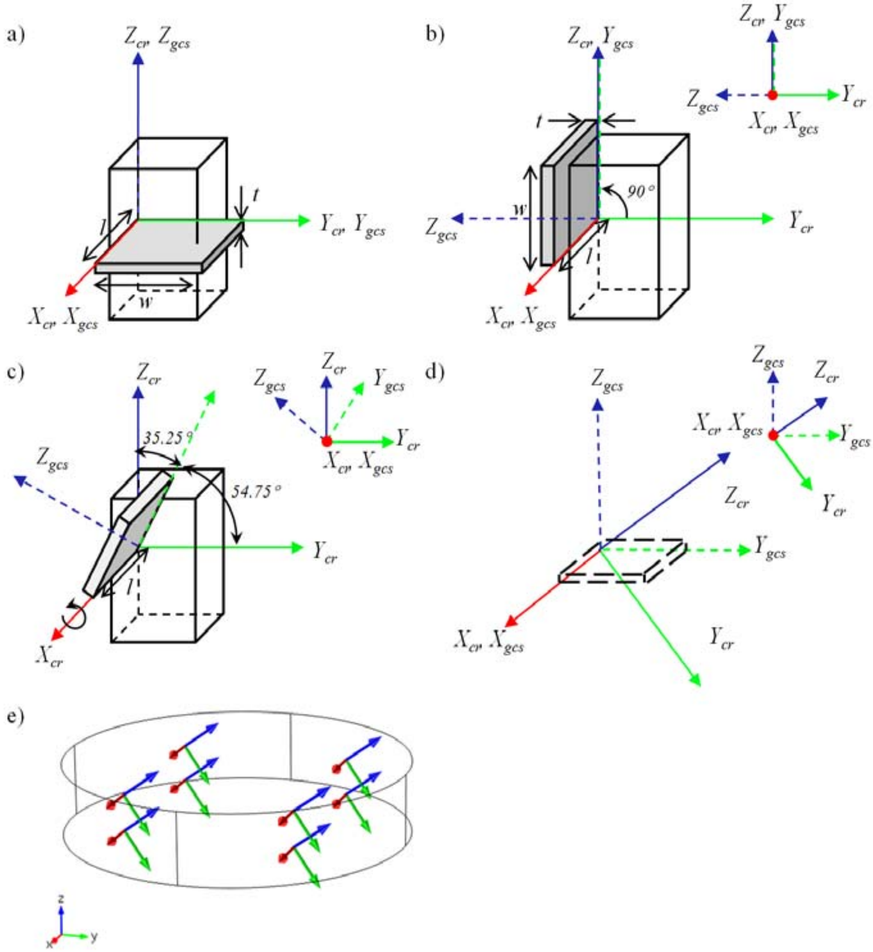


Figure 2-11: Defining an AT cut crystal plate within COMSOL, with normal in the global **Z***-direction. Within the 1978 IEEE standard the AT cut is defined as (YXl) −35.25°. Begin with the plate normal in the* Z_{cr}*-direction, so the crystal and global systems are coincident. Rotate the plate so that 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

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

and for the strain-charge formulation

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

where $\tilde{c}_E$, $\tilde{d}$, and ε 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 X is represented as pairs of a real-valued parameter

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

and a loss factor

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

the ratio of the imaginary and real part, and the complex data is then:

$$\tilde{X} = X(1 \pm j\eta_X)$$

where the sign depends on the material property used. The loss factors are specific to the material property, and thus they are named according to the property they refer to, for example, η_{cE}. For a structural material without coupling, simply use η_s, the structural loss factor.

The loss factors are defined so that a positive loss factor value 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

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

where m and n refer to components of each matrix.

Coupling Loss

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

Dielectric Loss

$$\tilde{\varepsilon}_{rS}^{m,n} = \varepsilon_{rS}^{m,n}(1-j\eta_{\varepsilon S}^{m,n})$$
$$\tilde{\varepsilon}_{rT}^{m,n} = \varepsilon_{rT}^{m,n}(1-j\eta_{\varepsilon T}^{m,n})$$

Note that the multiplication is applied component-wise.

In practice, it is often difficult to find complex-valued data for each of the matrix elements in the literature. The loss factors can also be entered as scalar isotropic factors independently of the material and the other coefficients.

For more information about hysteretic losses, see Ref. 1 to Ref. 4.

ELECTRICAL CONDUCTIVITY (TIME HARMONIC)

For frequency domain analyses, the electrical conductivity of the piezoelectric material (see Ref. 2, Ref. 5, and Ref. 6) can be defined. Depending on the formulation of the electrical equation, the electrical conductivity appears in the equation as an effective electric displacement

$$\nabla \cdot \left(\mathbf{D} - j\frac{\sigma_e \mathbf{E}}{\omega}\right) = 0$$

where σ_e is the material electrical conductivity, and $\mathbf{E}$ is the electric field. Note that the displacement current variables themselves do not contain any conductivity effects.

Both a dielectric loss factor and the electrical conductivity can be defined at the same time. 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).

The use of electrical conductivity in a damped eigenfrequency analysis leads to a nonlinear eigenvalue problem, which must be solved iteratively. To compute the correct eigenfrequency, run the eigenvalue solver once for a single mode. Then, set the computed solution to be the linearization point for the eigenvalue solver, which is defined in the Settings window for the Eigenvalue Solver node. Re-run the eigenvalue solver repeatedly until the solution no longer changes. This process must be repeated for each mode separately.

DIELECTRIC DISPERSION

When the Dielectric Loss subnode is used with the Dielectric Dispersion option, the following equations are solved in the time domain:

$$\nabla\cdot\left(\mathbf{D}+\tau_{\mathrm{d}}\frac{\partial\mathbf{D}}{\partial t}+\varepsilon_0\Delta\varepsilon_{\mathrm{r}S}\mathbf{E}\right) = 0$$

$$\mathbf{D} = e\varepsilon+\varepsilon_0\varepsilon_{\mathrm{r}S}\mathbf{E}$$

where two material parameters can be specified, the relaxation time τ_{d}, and the relative permittivity increment $\Delta\varepsilon_{\mathrm{r}S}$. 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. In the frequency domain, the time derivative is replaced by the factor $j\omega$, and the above equation can be rewritten as

$$\nabla\cdot\left[e\varepsilon+\varepsilon_0\left(\varepsilon_{\mathrm{r}S}+\frac{\Delta\varepsilon_{\mathrm{r}S}}{1+(\tau_{\mathrm{d}}\omega)^2}\right)\mathbf{E}-j\frac{\tau_{\mathrm{d}}\omega\varepsilon_0\Delta\varepsilon_{\mathrm{r}S}}{1+(\tau_{\mathrm{d}}\omega)^2}\mathbf{E}\right] = 0$$

which shows how the dispersion parameters contribute into the polarization and losses. Thus, the effective permittivity varies from $\varepsilon_{\mathrm{r}S}+\Delta\varepsilon_{\mathrm{r}S}$ down to $\varepsilon_0\varepsilon_{\mathrm{r}S}$ as the excitation frequency increases from zero. The damping effect vanishes for both large and small frequencies, and it reaches the maximum for $\omega = 1/\tau_{\mathrm{d}}$.

In the *COMSOL Multiphysics Reference Manual:*

- Selecting a Stationary, Time-Dependent, or Eigenvalue Solver
- Eigenvalue Solver

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** 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 non-solid, this is typically the case of air, in which the electrostatics is solved but not mechanics.
- Non-solid and non-electrostatics 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 rho (SI unit: kg/m^3)
- Elasticity matrix cE (SI unit: Pa) in Stress-charge form.
- Compliance Matrix sE (SI unit: 1/Pa) in Strain-charge form.

ELECTROSTATIC PROPERTIES:

- Relative Permittivity epsilonrS (dimensionless) in Stress-charge form.
- Relative Permittivity epsilonrT (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, eta_cE, or loss factor for elasticity matrix sE, eta_sE: required when Mechanical Damping is present. The latter is valid only in Strain-Charge form.
- Loss factor for electrical permittivity matrix εrS, eta_epsilonS, or Loss factor for electrical permittivity matrix εrT, eta_epsilonT: required when Dielectric Loss is present. The latter is valid only in Strain-Charge form.
- Loss factor for coupling matrix e, eta_eES, or loss factor for coupling matrix d, eta_dET: 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** 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** 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 non-solid, this is typically the case of air, in which the magnetics is solved but not mechanics.
- Non-solid and non-magnetics 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 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 ll 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:

Damping Sources

There are many sources of damping in a system. Some some them are:

- Dissipation in the material. This dissipation can be for example be proportional to the amplitude of the strain rate (viscous damping) or to the amplitude of the strain itself (hysteretic damping).
- Thermoelastic damping, which is a thermodynamic effect which is related not only to the state in a point, but also to the gradients of the temperature field.
- Damping caused by the surrounding medium, often air or water.
- Friction between joined parts.
- Components intended to supply damping like a dashpot in a car suspension.

It is often difficult to separate and quantify these effects, so damping modeling is one of the biggest challenges in structural dynamics.

About Damping

Phenomenological damping models are typically invoked to model the intrinsic frictional damping present in most materials (material damping). These models are easiest to understand in the context of a system with a single degree of freedom. The following equation of motion describes the dynamics of such a system with viscous damping:

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = f(t) \tag{2-16}$$

In this equation u is the displacement of the degree of freedom, m is its mass, c is the damping parameter, and k is the stiffness of the system. The time (t) dependent forcing term is $f(t)$. This equation is often written in the form:

$$\frac{d^2u}{dt^2} + 2\zeta\omega_0\frac{du}{dt} + \omega_0^2 u = \frac{f(t)}{m} \tag{2-17}$$

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. ζ can also be readily related to many of the various measures of damping employed in different disciplines. These are summarized in Table 2-7.

TABLE 2-7: RELATIONSHIPS BETWEEN MEASURES OF DAMPING

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

TABLE 2-7: RELATIONSHIPS BETWEEN MEASURES OF DAMPING

DAMPING PARAMETER	DEFINITION	RELATION TO DAMPING RATIO
Quality factor	$Q = \omega/\Delta\omega$ where $\Delta\omega$ is the bandwidth of the amplitude resonance measured at $1/\sqrt{2}$ of its peak.	$Q \approx 1/(2\zeta)$ $(\zeta \ll 1)$
Loss factor	$\eta = \frac{1}{2\pi}\left(\frac{Q_h}{W_h}\right)$ where Q_h is the energy lost per cycle and W_h is the maximum potential energy stored in the cycle. The variables Q_h and W_h are available as `solid.Qh` and `solid.Wh`.	At the resonant frequency: $\eta \approx 2\zeta$ $(\zeta \ll 1)$

In the frequency domain, the time dependence of the force and the displacement can be represented by introducing a complex force term and assuming a similar time dependence for the displacement. The equations

$$f(t) = \mathrm{Re}\{Fe^{j\omega t}\} \text{ and } u(t) = \mathrm{Re}\{Ue^{j\omega t}\}$$

are written where ω is the angular frequency and the amplitude terms U and F can in general be complex (the arguments provide information on the relative phase of signals). Usually the real part is taken as implicit and is subsequently dropped. Equation 2-16 takes the following form in the frequency domain:

$$-\omega^2 mU + j\omega cU + kU = F \tag{2-18}$$

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

$$-\omega^2 U + 2j\omega\omega_0\xi U + \omega_0^2 U = \frac{F}{m} \tag{2-19}$$

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 2-16. This means that the method can be applied generally in either the time or frequency domain. The parameter c in Equation 2-16 is defined as a fraction of the mass and the stiffness using two parameters, α_{dM} and β_{dK}, such that

$$c = \alpha_{dM} m + \beta_{dK} k \tag{2-20}$$

Substituting this relationship into Equation 2-16 and rearranging into the form of Equation 2-17 gives:

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

When there are many degrees of freedom m, k, and c become matrices and the technique can be generalized.

Rayleigh damping can therefore be identified as equivalent to the damping ratio at resonance of:

$$\zeta = \frac{1}{2}\left(\frac{\alpha_{dM}}{\omega_0} + \beta_{dK}\omega_0\right) \tag{2-21}$$

Note that Equation 2-21 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 2-18). 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 2-21, 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 2-12 shows.

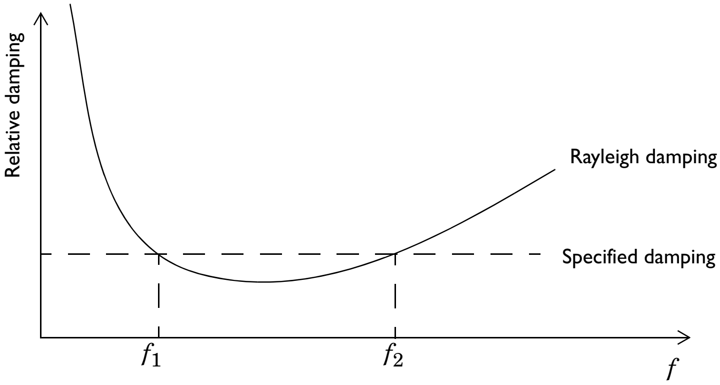


Figure 2-12: An example of Rayleigh damping.

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

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

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 2-21, a damping which increases linearly with frequency is obtained. If the damping ratio $\xi(f_0)$ 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 $\boldsymbol{D}^{\mathrm{c}}$

$$D^{\mathrm{c}} = (1 + j\eta_{\mathrm{s}})D$$

For a nonlinear elastic material, this applies to the tangential stiffness.

The use of loss factor damping traditionally refers to a scalar-valued loss factor η_{s}. But there is no reason that η_{s} must be scalar. Because the loss factor is a value deduced from true complex-valued material data, it can be represented by a matrix of the same dimensions as the anisotropic stiffness matrix. Especially for orthotropic materials, there should be a set of loss factors of all normal and shear elasticity modulus components, and COMSOL allows all these options, so a more general expression is.

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

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

$$S = (1+j\eta_{\mathrm{s}})\frac{\partial W_s}{\partial E}$$

Loss factor damping is available for *frequency response* analysis and damped eigenfrequency analysis in all interfaces.

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

$$G^* = G' + jG'' = (1+j\eta_{\mathrm{s}})G'$$

where G' is the storage modulus, G'' is the loss modulus, and their ratio $\eta_{\mathrm{s}} = G''/G'$ 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.

Explicit Damping

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.

Equivalent Viscous Damping

Although equivalent viscous damping is independent of frequency, it is only possible to use it in a frequency response analysis. Equivalent viscous damping also uses a loss factor as the damping parameter, and can be implemented using the Rayleigh damping feature, by setting 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.

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 2-20. 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
- The Fluid-Structure Interaction Interface
- 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 2-13). In this case the deformation of the beam introduces an effect known as *geometric nonlinearity* into the equations of solid mechanics.

Figure 2-13 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 non-orthogonality 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 2-13) 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 2-13 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.

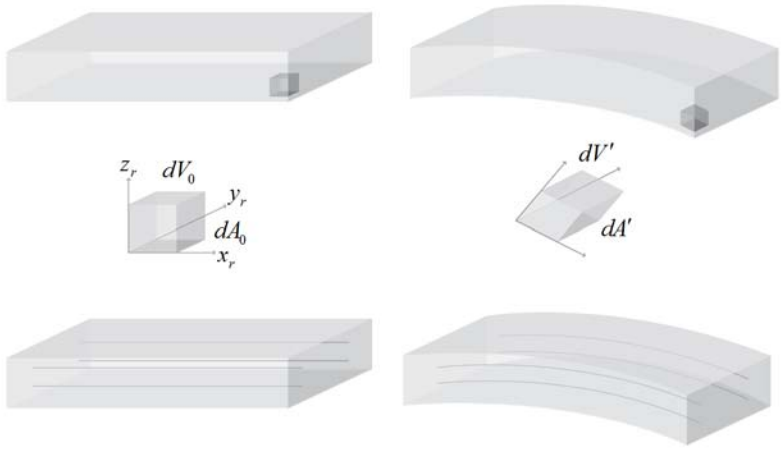


Figure 2-13: An example of the deformation of a beam showing the undeformed state (left) and the deformed state (right) of the beam itself with an element near its tip highlighted (top), of the element (center) and of lines parallel to the x-axis in the undeformed state (bottom).

It is important to note that, as the solid deforms, the Lagrangian frame becomes a non-orthogonal curvilinear coordinate system (see the lower part of Figure 2-13 to see the deformation of the *X*-axis). Particular care is therefore required when defining physics in this coordinate system.

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.

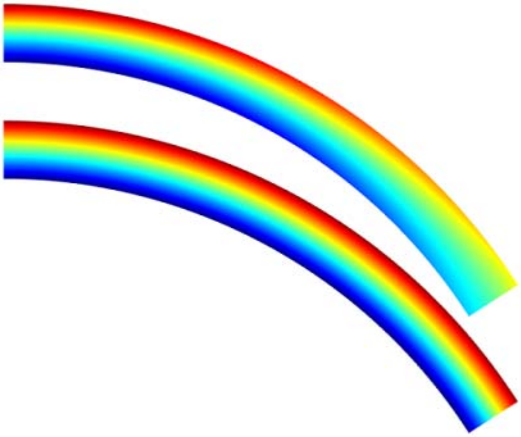


Figure 2-14: xx-components of the Cauchy stress tensor (top) and Second-Piola-Kirchoff stress tensor (below) for an initially straight beam

Another stress measure available in COMSOL 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. 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.

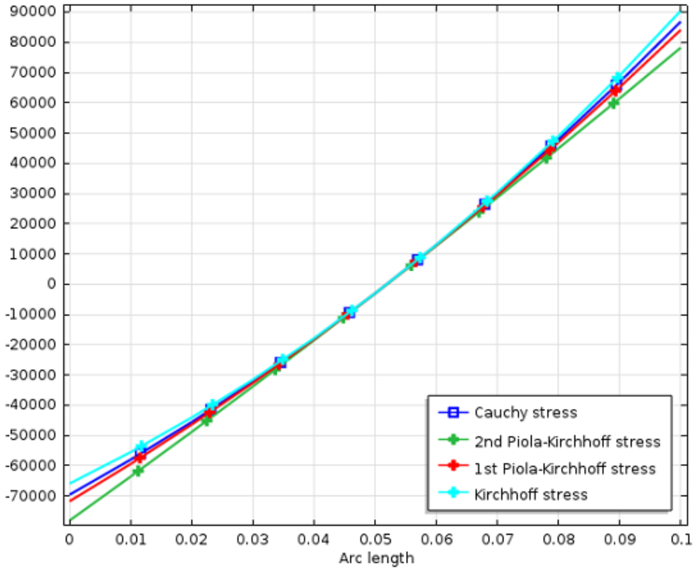


Figure 2-15: Stress distribution across the beam at the constrained end

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, a separate equation is solved to produce this mapping—defined by the mesh smoothing method (Laplacian, Winslow, hyperelastic, or Yeoh) with boundary conditions that determine the limits of deformation (these are usually determined by the physics of the system, whilst the domain level equations are typically being defined for numerical convenience). The ALE method offers significant advantages since the physical equations describing the system can be solved in a moving domain.

Deformed Geometry and Moving Mesh in the *COMSOL Multiphysics Reference Manual*

Geometric Nonlinearity for the Solid Mechanics Interface

For the Solid Mechanics interface, or any multiphysics interface derived from it, 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

- The Fluid-Structure Interaction interface.
- 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 or frequency domain 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 effects of the preload can be computed with or without taking geometric nonlinearity into account. In the second study step, where the 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.

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.

CONTACT AND FREQUENCY DOMAIN ANALYSIS

If a contact is included in the prestress load case, you can perform a subsequent frequency domain analysis in which then the linearization will be made around the computed contact state. It is however not possible perform an eigenvalue analysis based on a contact solution.

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: **Prestressed Analysis, Eigenfrequency**; **Prestressed Analysis, Frequency Domain**; 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** or **External Stress** node. Then add a separate **Eigenfrequency** study and select **Include Geometric Nonlinearity** in the settings for the study step.

Prestressed Analysis, Eigenfrequency, Prestressed Analysis, Frequency Domain, 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}:

  $$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \sum_k \frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}\right) \tag{2-22}$$

  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:

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

where S is the second Piola-Kirchhoff stress; ε is the Green-Lagrange strain, $\mathbf{E}_\mathrm{m}$ and $\mathbf{P}_\mathrm{m}$ are the electric field and electric polarization in the material orientation; $\mathbf{I}$ is the identity matrix; and c_E, e, and $\varepsilon_{\mathrm{r}S}$ are the piezoelectric material constants. The expression within parentheses equals the dielectric susceptibility of the solid:

$$\chi = \varepsilon_{\mathrm{r}S} - \mathbf{I}$$

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

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

where C is the right Cauchy-Green tensor

$$C = F^T F$$

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

$$\begin{aligned}\mathbf{E} &= F^{-T}\mathbf{E}_{\mathrm{m}}\\ \mathbf{P} &= J^{-1}F\mathbf{P}_{\mathrm{m}}\\ \mathbf{D} &= J^{-1}F\mathbf{D}_{\mathrm{m}}\\ \rho_{\mathrm{v}} &= \rho_{\mathrm{V}}J^{-1}\end{aligned} \qquad (2\text{-}23)$$

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

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

Finally, one can rewrite the constitutive equations as

$$\begin{aligned}S &= c_E\varepsilon - e^T\mathbf{E}_{\mathrm{m}}\\ \mathbf{D}_{\mathrm{m}} &= e\varepsilon + \varepsilon_0\varepsilon_{\mathrm{r}S}\mathbf{E}_{\mathrm{m}} + \varepsilon_0(JC^{-1} - \mathbf{I})\mathbf{E}_{\mathrm{m}}\end{aligned}$$

DECOUPLED MATERIALS WITH LARGE DEFORMATIONS

The large deformation formulation described in the previous section applies directly to 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 non-solid 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 non-solid 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

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.

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.

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 (`Eequiv`) of the material on the destination side. For linear elastic isotropic materials `Eequiv` is equal to the actual Young’s modulus. For other types of materials `Eequiv` 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 in 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 on 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 it 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 ab 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 if 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 are 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 2-8 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 2-8: 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 2-8: 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	Non-zero 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	Non-zero 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	Non-zero 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	Non-zero 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 2-8: 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 n this section you will find a summary of variables created in the contact feature which can be useful in postprocessing.I

TABLE 2-9: 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 2-9: 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**	ntegration over all contact boundaries

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

$$\mathbf{f}_{\mathrm{sl}} = (1 + i\eta)\mathbf{f}_{\mathrm{s}}$$

where $\mathbf{f}_{\mathrm{s}}$ 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-Structural Interaction
- Acoustic-Structure Interaction
- Thermal-Electric-Structural Interaction

Thermal-Structural 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 Inputs 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 Inputs section on the Settings window. Some of these sections may be empty, specifically 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.

DENSITY

All Structural Mechanics interfaces are formulated on the material frame. This means that the equations of motion are formulated 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 you must not use a temperature dependent density. 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 is represented a *thermal strain* $\varepsilon_{\mathrm{th}}$, which produces stress-free deformations. For a linear elastic material, the constitutive law is

$$\sigma = C : (\varepsilon - \varepsilon_{\mathrm{th}})$$

In the computations, the thermal expansion appears as a load, even though it formally is a part of the constitutive relations.

- You can include thermal expansion in a model either by adding a Thermal Expansion (for materials) subnode to the chosen material, or by using the Thermal Expansion (Multiphysics Coupling) and Temperature Coupling (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 The Thermal Stress Interface or the 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

$$\frac{dL}{L} = \alpha_t(T)dT \tag{2-24}$$

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, since 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:

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

Converting from Tangent Form to Secant Form

Equation 2-24 can be integrated, giving

$$\ln\left(\frac{L}{L_0}\right) = \int_{T_{\mathrm{ref}}}^{T} \alpha_{\mathrm{t}}(\tau)d\tau = I(T,T_{\mathrm{ref}}) \qquad (2\text{-}25)$$

Thus,

$$\frac{\Delta L}{L_0} = \frac{L}{L_0}-1 = e^{I(T,T_{\mathrm{ref}})}-1$$

giving the secant thermal expansion coefficient as

$$\alpha(T,T_{\mathrm{ref}}) = \frac{e^{I(T,T_{\mathrm{ref}})}-1}{(T-T_{\mathrm{ref}})}$$

For most materials and temperature ranges $I(T,T_{\mathrm{ref}}) \ll 1$, which makes it possible to approximate with the simpler expression

$$\alpha(T,T_{\mathrm{ref}}) = \frac{I(T,T_{\mathrm{ref}})}{(T-T_{\mathrm{ref}})} \qquad (2\text{-}26)$$

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 2-25 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.
- Pre-compute the expression in Equation 2-24 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.

Thermal Expansion Coefficient Dependence on Reference Temperature

Let $\alpha_{\mathrm{m}}(T)$ 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 $\alpha_{\mathrm{m}}(T)$ is concerned. Denote the length of the sample at a temperature T as $L(T)$ and the strain-free length as $L_0 = L(T_{\mathrm{m}})$. The dilation can be then expressed as $L(T) - L(T_{\mathrm{m}})$. Using the definition of the secant coefficient of thermal expansion, $L(T)$ can be written as:

$$L(T) = [1 + \alpha_{\mathrm{m}}(T)(T - T_{\mathrm{m}})]L(T_{\mathrm{m}}) \tag{2-27}$$

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 $L(T) - L(T_{\mathrm{ref}})$, where $L(T_{\mathrm{ref}})$ can be written as.

$$L(T_{\mathrm{ref}}) = [1 + \alpha_{\mathrm{m}}(T_{\mathrm{ref}})(T_{\mathrm{ref}} - T_{\mathrm{m}})]L(T_{\mathrm{m}}) \tag{2-28}$$

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 $L(T_{\mathrm{ref}})$ can be related using Equation 2-27 but with T_{m} replaced by T_{ref}.

$$L(T) = [1 + \alpha_{\mathrm{r}}(T)(T - (T_{\mathrm{ref}}))]L((T_{\mathrm{ref}})) \tag{2-29}$$

Here $\alpha_{\mathrm{r}}(T)$ is the redefined thermal expansion coefficient, based on T_{ref}. It can be derived from the relations above. Using Equation 2-27 and Equation 2-29 there are two ways of writing the current length $L(T)$, so that

$$[1+\alpha_{\mathrm{r}}(T)(T-(T_{\mathrm{ref}}))]L((T_{\mathrm{ref}})) = [1+\alpha_{\mathrm{m}}(T)(T-T_{\mathrm{m}})]L(T_{\mathrm{m}}) \qquad (2\text{-}30)$$

Equation 2-28 makes it is possible to eliminate $L(T_{\mathrm{ref}})$ and $L(T_{\mathrm{m}})$ from Equation 2-30, and after some algebra $\alpha_{\mathrm{r}}(T)$ can then be written as

$$\alpha_{\mathrm{r}}(T) = \frac{\alpha_{\mathrm{m}}(T) + (T_{\mathrm{ref}} - T_{\mathrm{m}})\dfrac{\alpha_{\mathrm{m}}(T) - \alpha_{\mathrm{m}}(T_{\mathrm{ref}})}{T - T_{\mathrm{ref}}}}{1 + \alpha_{\mathrm{m}}(T_{\mathrm{ref}})(T_{\mathrm{ref}} - T_{\mathrm{m}})} \qquad (2\text{-}31)$$

Representation in COMSOL

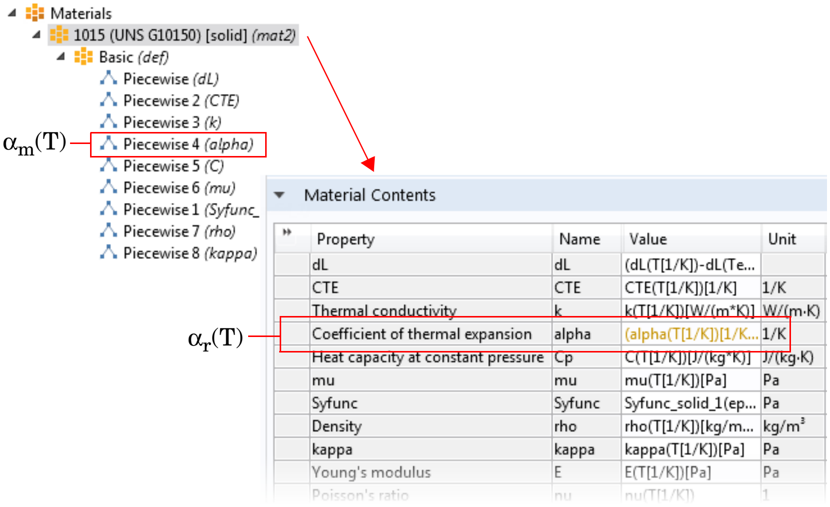
Most materials listed in COMSOL's material libraries and databases contain a function for the measured temperature-dependent thermal expansion coefficient curve. You can find this from the Materials branch, as shown in Figure 2-16. The Piecewise function named `alpha_solid_1` is the measured thermal expansion coefficient $\alpha_{\mathrm{m}}(T)$.

Using Functions in Materials in the *COMSOL Multiphysics Reference Manual*

The Material Contents section in Figure 2-16 shows the material property `alpha`, which is the redefined thermal expansion coefficient $\alpha_{\mathrm{r}}(T)$. 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 2-31, but with a small modification to avoid problems if $T{=}T_{\mathrm{ref}}$.



Property	Name	Value	Unit
dL	dL	(dL(T[1/K])-dL(Te...	
CTE	CTE	CTE(T[1/K])[1/K]	1/K
Thermal conductivity	k	k(T[1/K])[W/(m*K)]	W/(m·K)
Coefficient of thermal expansion	alpha	(alpha(T[1/K])[1/K...	1/K
Heat capacity at constant pressure	Cp	C(T[1/K])[J/(kg*K)]	J/(kg·K)
mu	mu	mu(T[1/K])[Pa]	Pa
Syfunc	Syfunc	Syfunc_solid_1(ep...	Pa
Density	rho	rho(T[1/K])[kg/m...	kg/m³
kappa	kappa	kappa(T[1/K])[Pa]	Pa
Young's modulus	E	E(T[1/K])[Pa]	Pa
Poisson's ratio	nu	nu(T[1/K])	1

Figure 2-16: 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, 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).

- 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.

User-Defined Materials and Libraries in the *COMSOL Multiphysics Reference Manual*

Constraints and Thermal Expansion

When a structure is undergoes thermal expansion, the fact that the motion is restricted by constraints will in general cause stresses. There are three types of such effects:

- The global expansion is restricted by constraints at different locations, so that internal forces are introduced throughout the structure.
- At a boundary 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 relive the stress at constraints in a heated structure, see *Thermal Expansion in a MEMS Device*: Application Library path **Structural_Mechanics_Module/Thermal-Structure Interaction/ thermal_expansion**.

Thermoelastic Damping

In most engineering problems, the coupling between temperatures and structural problems can be considered as unidirectional. Only the thermal expansion is 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 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.

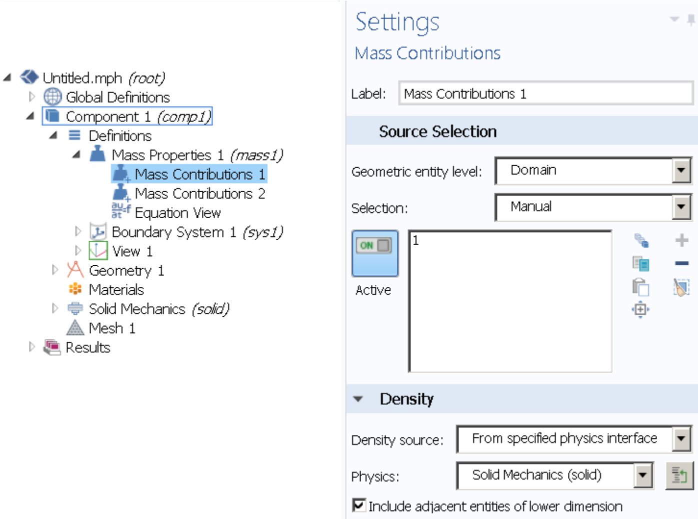
In either case, the heat source term is only present when Structural Transient Behavior is set to **Include inertial terms**.

See also

- Entropy and Thermoelasticity
- Thermal Expansion (for materials)
- Thermal Expansion (Multiphysics Coupling)

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.



Settings

Mass Contributions

Label: Mass Contributions 1

Source Selection

Geometric entity level: Domain

Selection: Manual

1

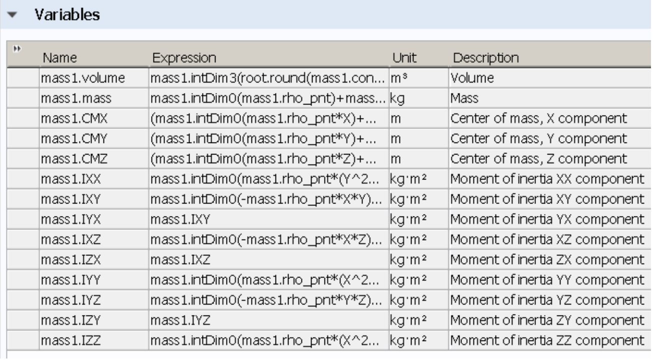
Density

Density source: From specified physics interface

Physics: Solid Mechanics (solid)

☑ Include adjacent entities of lower dimension

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

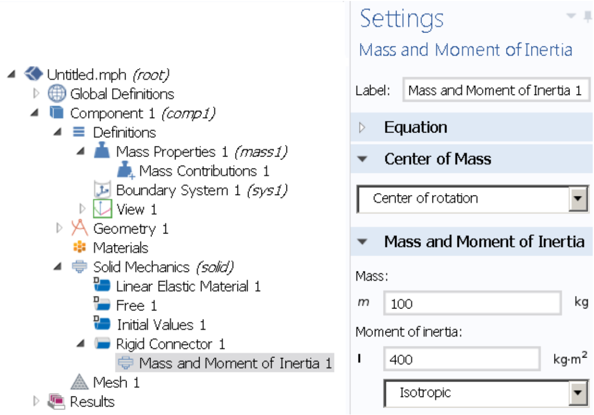
Name	Expression	Unit	Description
mass1.volume	mass1.intDim3(root.round(mass1.con...	m^3	Volume
mass1.mass	mass1.intDim0(mass1.rho_pnt)+mass...	kg	Mass
mass1.CMX	(mass1.intDim0(mass1.rho_pnt*X)+...	m	Center of mass, X component
mass1.CMY	(mass1.intDim0(mass1.rho_pnt*Y)+...	m	Center of mass, Y component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



Untitled.mph *(root)*
- Global Definitions
- Component 1 *(comp1)*
  - Definitions
    - Mass Properties 1 *(mass1)*
      - Mass Contributions 1
    - Boundary System 1 *(sys1)*
    - View 1
  - Geometry 1
  - Materials
  - Solid Mechanics *(solid)*
    - Linear Elastic Material 1
    - Free 1
    - Initial Values 1
    - Rigid Connector 1
      - Mass and Moment of Inertia 1
  - Mesh 1
- Results

Settings

Mass and Moment of Inertia

Label: Mass and Moment of Inertia 1

▷ Equation

▼ Center of Mass

Center of rotation

▼ Mass and Moment of Inertia

Mass:

m 100 kg

Moment of inertia:

I 400 kg·m^2

Isotropic

- 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 inertial 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 chose **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 and Tables

Pre-tensioned Bolts

The information about pre-tensioned 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 pre-stress in the bolt must be taken into account in order to correctly capture the behavior under service loads. The Bolt Pre-Tension functionality in COMSOL Multiphysics designed to simplify such analyses.

During mounting, a bolt is tightened to a certain pre-stress. 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. Usually it is most efficient to add the predefined bolt geometries from the Part Libraries.
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 2-17). 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 `Pre-tension 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 2-18).
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 Pre-Tension** node, in which the pre-tension force or stress is prescribed for a set of bolts with the same data.

6 For each bolt having the identical pre-tension 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 pre-tension 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 Pre-Tension** node for a set of similar bolts where some of them are located in symmetry planes.

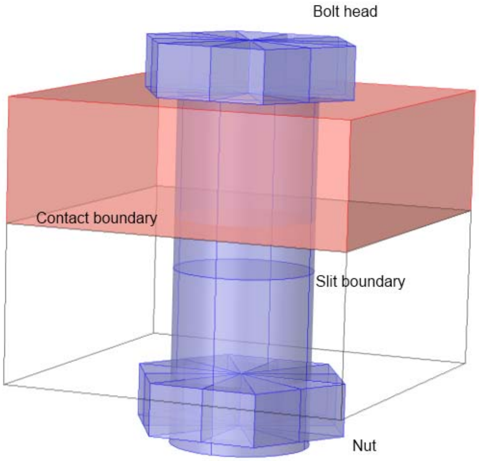
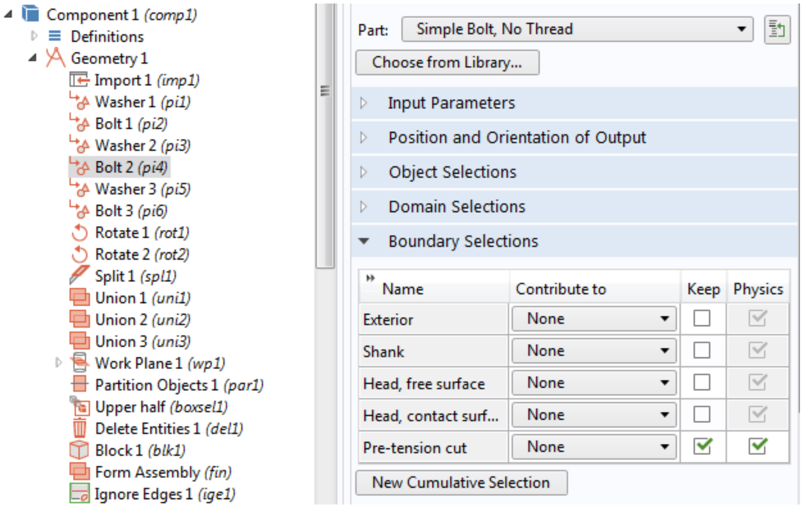


Figure 2-17: Example of a bolted joint.



Part: Simple Bolt, No Thread

Name	Contribute to	Keep	Physics
Exterior	None	□	☑
Shank	None	□	☑
Head, free surface	None	□	☑
Head, contact surf...	None	□	☑
Pre-tension cut	None	☑	☑

Figure 2-18: 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:

$$\mathbf{u}_\mathrm{u} = \mathbf{u}_\mathrm{d} - d\mathbf{n}$$

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.

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.

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 pre-stress 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 Pre-Tension** is designed for this.

2 Add one or more studies or study steps to analyze the effects of the service loads.

3 Since the pre-tension degrees of freedom are not solved for in the service load study steps, they must obtain their values from the pre-tension 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 pre-tension study step.

It is important to make sure that you only solve for the bolt predeformation degrees of freedom in the pre-tension study step, and not when analyzing the service loads.

If you use the **Bolt Pre-Tension** study type for the pre-tension study step, and any other study type to analyze the service loads, the solvers are automatically set up to handle this. The **Bolt Pre-Tension** 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 Pre-Tension tag>.<Bolt Selection tag>.<variable>`

must be used. An example could be `solid.pblt1.sblt1.F_bolt`. The bolt results are summarized in the table below.

TABLE 2-10: BOLT VARIABLES

VARIABLE	DESCRIPTION
`d_pre`	Predeformation
`F_bolt`	Axial force in the bolt
`F_shear`	Shear force in the bolt

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**

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 is 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, $x1$-$x2$-$x3$. The $x1$ 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 $x2$ direction, and thus implicitly the x3 direction. You specify the orientation either by selecting a point in the $x1$-$x2$ plane, or by defining an orientation vector in an approximate $x2$ direction. In either case, the actual $x2$ direction is chosen so that it is perpendicular to the SCL, and lies in the plane you have specified. The $x3$ orientation is then taken as perpendicular to $x1$ and $x2$. As long as you are only interested in a stress intensity, the choice of orientation is arbitrary.
- In 2D, the $x3$ direction is the out-of-plane direction, and the $x2$ direction is perpendicular to the SCL in the X-Y plane.
- In 2D axial symmetry, the $x3$ direction is the azimuthal direction, and the $x2$ 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.

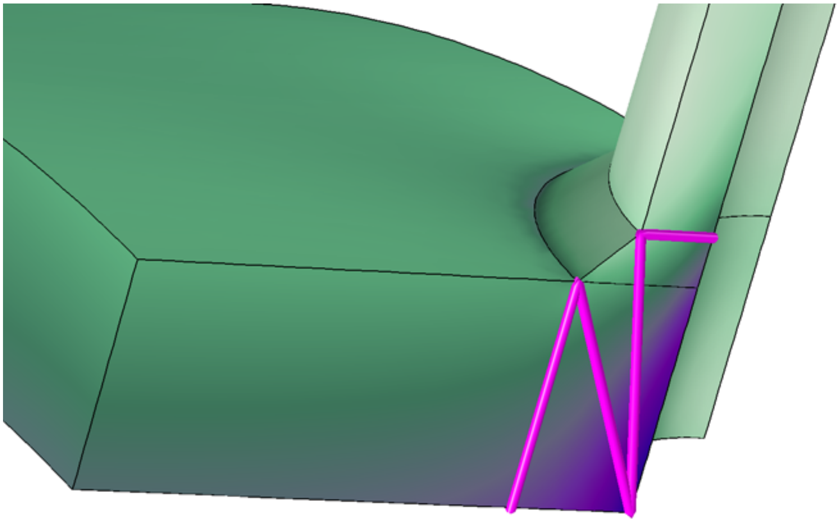


Figure 2-19: Four stress classification lines in a transition region at a pressure vessel nozzle.

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 re-run the analysis, but this may not be a good solutions 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 have displacements controlled by the solution from the larger model.

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

1 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.

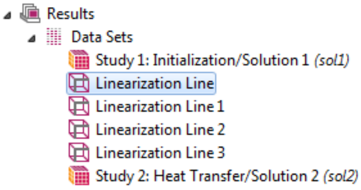
STUDIES AND SOLUTIONS

Stress linearization is a pure postprocessing operation. The a **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.



- Results
  - Data Sets
    - Study 1: Initialization/Solution 1 *(sol1)*
    - Linearization Line
    - Linearization Line 1
    - Linearization Line 2
    - Linearization Line 3
    - Study 2: Heat Transfer/Solution 2 *(sol2)*

Figure 2-20: 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.

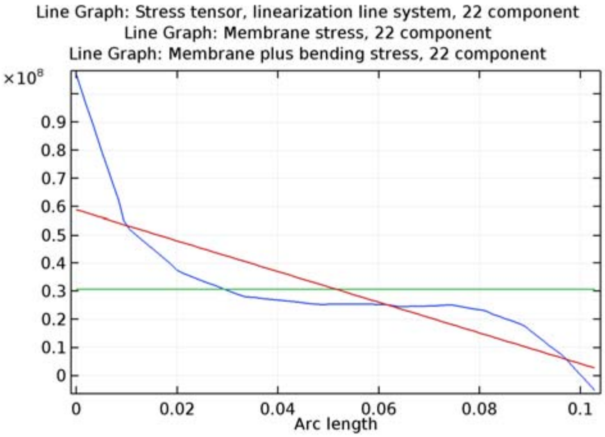


Figure 2-21: Default plot along a stress classification line.

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 `'sl1'`.

As an example, the variable `solid.Sm22` and the variable `solid.sl1.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 2-11: VARIABLE FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
Sl*ij*	Stress tensor in local coordinate system	σ_{ij}	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Sm*ij*	Membrane stress	$\sigma_{\mathrm{m},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Sbmax*ij*	Maximum bending stress	$\sigma_{\mathrm{b(max)},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
Sb*ij*	Bending Stress	$\sigma_{\mathrm{b},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Smb*ij*	Membrane + bending stress	$\sigma_{\mathrm{mb},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Physics, Feature
Sps*ij*	Peak stress, start point	$\sigma_{\mathrm{p(start)},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
Spe*ij*	Peak stress, start point	$\sigma_{\mathrm{p(end)},\,ij}$	*ij* = 11, 12, 13, 22, 23, 33	Feature
SIm	Stress intensity, membrane	σ_{int}		Physics, Feature
SImbs	Stress intensity, membrane + bending, start point	σ_{int}		Feature
SIbme	Stress intensity, membrane + bending, end point	σ_{int}		Feature

TABLE 2-11: VARIABLE FOR STRESS LINEARIZATION

VARIABLE	DESCRIPTION	VARIABLE IN THEORY SECTION	COMMENT	SCOPE
SImb	Stress intensity, membrane + bending,		max(SImbs, SImbe)	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 with only a weak unsymmetry, 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 presmoother and postsmoother 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

Viscoelastivity, 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 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 Pre-tensioned Bolts functionality.
- 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.

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*

3

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

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

$$\sigma = \mathsf{C}:\varepsilon$$

Here, the stress tensor σ and the strain tensor ε are second-order tensors, while the constitutive tensor C is a fourth-order tensor. The ':' symbol means a contraction over two indices. In a notation where the indices are shown, the same equation would read

$$\sigma_{ij} = \mathsf{C}_{ijkl}\varepsilon_{kl}$$

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

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

In a few cases, 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.

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

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

in space. Because the material coordinates are constant, the current spatial position is uniquely determined by the displacement vector $\mathbf{u}$, pointing from the reference position to the current position. The global Cartesian components of this displacement vector in the spatial frame, by default called u, v, and w, are the primary dependent variables in the Solid Mechanics interface.

By default, the Solid Mechanics interface uses the calculated displacement and Equation 3-1 to define the difference between spatial coordinates $\mathbf{x}$ and material coordinates $\mathbf{X}$. 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
- Out-of-plane and Circumferential Modes

Lagrangian Formulation

The formulation used for structural analysis in COMSOL Multiphysics for both small and finite deformations is a *total Lagrangian* formulation. This means that the computed stress and deformation state is always referred to the material configuration rather than to current position in space.

Likewise, material properties are always given for material particles and with tensor components referring to a coordinate system based on the material frame. This has the obvious advantage that spatially varying material properties can be evaluated just once for the initial material configuration, and they do not change as the solid deforms and rotates.

Consider a certain physical particle, initially located at the coordinate $\mathbf{X}$. During deformation, this particle follows a path

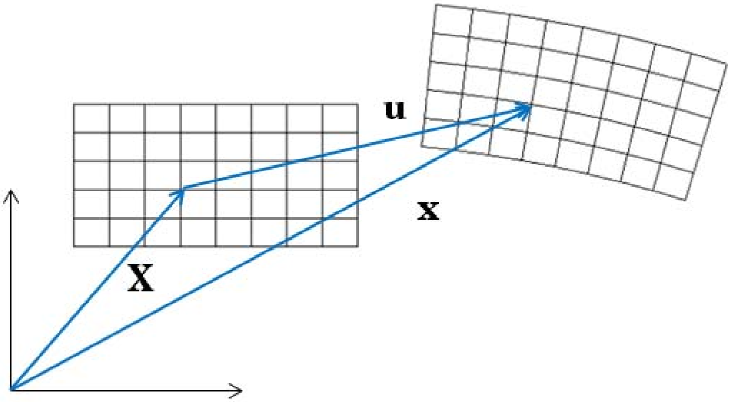
$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$$

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

For simplicity, assume that undeformed and deformed positions are measured in the same coordinate system. Using the displacement $\mathbf{u}$ it is then possible to write

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

The displacement is considered as a function of the material coordinates (X, Y, Z), but it is not explicitly a function of the spatial coordinates (x, y, z). It is thus only possible to compute derivatives with respect to the material coordinates.



In the following, the gradient operator is assumed to be a gradient with respect to the material coordinates, unless something else is explicitly stated.

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

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

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

The *deformation gradient* tensor $\boldsymbol{F}$ shows how an infinitesimal line element, $d\mathbf{X}$, is mapped to the corresponding deformed line element $d\mathbf{x}$ by

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

The deformation gradient $\boldsymbol{F}$ contains the complete information about the local straining and rotation of the material. It is a two-point tensor (or a double vector), which transforms as a vector with respect to each of its indices. It involves both the reference and present configurations.

In terms of the displacement gradient, $\boldsymbol{F}$ can be written as

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

The deformation of the material (stretching) will in general cause changes in the material density. The ratio between current and initial volume (or mass density) is given by

$$\frac{\mathrm{d}V}{\mathrm{d}V_0} = \frac{\rho_0}{\rho} = \det(F) = J$$

Here, ρ_0 is the initial density and ρ is the current density after deformation. The determinant of the deformation gradient tensor $\boldsymbol{F}$ is related to volumetric changes with respect to the initial state. A pure rigid body displacement implies $J = 1$. Also, an incompressible material is represented by $J = 1$. These are called *isochoric* processes.

The determinant of the deformation gradient tensor is always positive (since a negative mass density is unphysical). The relation $\rho = \rho_0 / J$ implies that for $J < 1$ there is compression, and for $J > 1$ there is expansion. Since $J > 0$, the deformation gradient $\boldsymbol{F}$ is invertible.

In the material formulations used within the structural mechanics interfaces, the mass density should in general be constant because the equations are formulated for fixed material particles. 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 $\mathbf{U}$ tensor are the principal stretches, λ_1, λ_2, and λ_3. The stretch of a line element with initial length L_0 and current length L is

$$\lambda = \frac{L}{L_0} = 1 + \varepsilon_{\mathrm{eng}}$$

where $\varepsilon_{\mathrm{eng}}$ is the engineering strain. The three principal stretches act along three orthogonal directions. In the coordinate system defined by these principal directions, the $\mathbf{U}$ tensor will be diagonal:

$$U = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

The *right Cauchy-Green* deformation tensor C defined by

$$C = F^T F = U^2$$

It is a symmetric and positive definite tensor, which accounts for the strain but not for the rotation. The eigenvalues of the C tensor are the squared principal stretches, thus providing a more efficient way to compute the principal stretches than by using the stretch tensor U directly.

The *Green-Lagrange* strain tensor is a symmetric tensor defined as

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

Since C is independent of rigid body rotations, this applies also to the Green-Lagrange strain tensor.

Using the displacement components and Cartesian coordinates, the Green-Lagrange strain tensor can be written on component form as

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

The rotation independence of the Green-Lagrange strain tensor, together with the fact that it for small strain approaches the engineering strain tensor explains why it is a common choice in constitutive models for small strain- finite rotation. As an opposite, a pure rigid rotation causes strains when engineering strains are used.

The Green-Lagrange is 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.el11`, `solid.el12`, and so on.
- In a geometrically linear analysis, the strain variables `solid.eX,solid.el11`, and so on, will instead represent engineering strain.
- The right Cauchy-Green deformation tensor in local coordinate system are named `solid.Cl11`, `solid.Cl12`, and so on.

Some textbooks prefer to use the *left Cauchy-Green* deformation tensor $B = FF^T$, which is also symmetric and positive definite but it is defined in the spatial frame.

ENGINEERING STRAIN

Under the assumption of small displacements and rotations, the normal strain components and the shear strain components are related to the deformation as follows:

$$
\begin{aligned}
\varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_{xy} &= \frac{\gamma_{xy}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\varepsilon_y &= \frac{\partial v}{\partial y} & \varepsilon_{yz} &= \frac{\gamma_{yz}}{2} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \qquad (3\text{-}3) \\
\varepsilon_z &= \frac{\partial w}{\partial z} & \varepsilon_{xz} &= \frac{\gamma_{xz}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right).
\end{aligned}
$$

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

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

A 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

$$(\varepsilon - \varepsilon_{\mathrm{p}} I)\mathbf{v}_{\mathrm{p}} = 0$$

The three principal strains are sorted so that

$$\varepsilon_{\mathrm{p}1} \geq \varepsilon_{\mathrm{p}2} \geq \varepsilon_{\mathrm{p}3}$$

This sorting is true also for the 2D cases. The corresponding vectors in the principal directions, $\mathbf{v}_{\mathrm{p}i}$, 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:

$$\lambda_{\mathrm{p1}} \ge \lambda_{\mathrm{p2}} \ge \lambda_{\mathrm{p3}}$$

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

$$I_1(\varepsilon) = \mathrm{trace}(\varepsilon)$$
$$I_2(\varepsilon) = \frac{1}{2}(\mathrm{trace}(\varepsilon)^2 - \mathrm{trace}(\varepsilon^2))$$
$$I_3(\varepsilon) = \det(\varepsilon)$$

The invariants of the strain deviator tensor is also useful.

$$\mathrm{dev}(\varepsilon) = \varepsilon - \frac{1}{3}\mathrm{trace}(\varepsilon)I$$
$$J_1(\varepsilon) = \mathrm{trace}(\mathrm{dev}(\varepsilon)) = 0$$
$$J_2(\varepsilon) = \frac{1}{2}\mathrm{dev}(\varepsilon):\mathrm{dev}(\varepsilon) = \frac{1}{3}I_1(\varepsilon)^2 - I_2(\varepsilon)$$
$$J_3(\varepsilon) = \det(\mathrm{dev}(\varepsilon)) = \frac{2}{27}I_1(\varepsilon)^3 - \frac{1}{3}I_1(\varepsilon)I_2(\varepsilon) + I_3(\varepsilon)$$

As defined above $J_2 \ge 0$. I_1 represents the relative change in volume for infinitesimal strains and J_2 represents the magnitude of shear strain.

In tensor component notation, the invariants can be written as

$$\mathrm{trace}(\varepsilon) = \varepsilon_{kk}$$

$$\mathrm{dev}(\varepsilon)_{ij} = \varepsilon_{ij} - \frac{\varepsilon_{kk}}{3}\delta_{ij}$$

$$I_2(\varepsilon) = \frac{1}{2}(\varepsilon_{ii}\varepsilon_{jj} - \varepsilon_{ij}\varepsilon_{ji})$$

$$J_2(\varepsilon) = \frac{1}{2}\mathrm{dev}(\varepsilon)_{ij}\mathrm{dev}(\varepsilon)_{ji}$$

The volumetric strain is defined as

$$\varepsilon_{\mathrm{vol}} = \mathrm{trace}(\varepsilon)\,.$$

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

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

$$I_1(\varepsilon) = \varepsilon_{\mathrm{p}1} + \varepsilon_{\mathrm{p}2} + \varepsilon_{\mathrm{p}3}$$

$$I_2(\varepsilon) = \varepsilon_{\mathrm{p}1}\varepsilon_{\mathrm{p}2} + \varepsilon_{\mathrm{p}2}\varepsilon_{\mathrm{p}3} + \varepsilon_{\mathrm{p}1}\varepsilon_{\mathrm{p}3}$$

$$I_3(\varepsilon) = \varepsilon_{\mathrm{p}1}\varepsilon_{\mathrm{p}2}\varepsilon_{\mathrm{p}3}$$

The principal stresses are the roots of the *characteristic equation* (Cayley–Hamilton theorem)

$$\varepsilon_{\mathrm{p}}^{3} - I_1\varepsilon_{\mathrm{p}}^{2} + I_2\varepsilon_{\mathrm{p}} - I_3 = 0$$

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

$$L = L_\mathrm{d} + L_\mathrm{w}$$

where

$$L_\mathrm{d} = \frac{1}{2}(L + L^T)$$

is called the *rate of strain tensor*, and

$$L_\mathrm{w} = \frac{1}{2}(L - L^T)$$

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

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

The spin tensor $L_\mathrm{w}(\mathbf{x},t)$ 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, $\varepsilon_{\mathrm{th}}$
- Hygroscopic strain, $\varepsilon_{\mathrm{hs}}$
- Plastic strain, $\varepsilon_{\mathrm{pl}}$
- Creep strain, $\varepsilon_{\mathrm{cr}}$
- Viscoplastic strain, $\varepsilon_{\mathrm{vp}}$

ADDITIVE DECOMPOSITION

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

$$\varepsilon_{\mathrm{e}} = \varepsilon - \varepsilon_{\mathrm{inel}}$$

where

$$\varepsilon_{\mathrm{inel}} = \varepsilon_0 + \varepsilon_{\mathrm{ex}} + \varepsilon_{\mathrm{th}} + \varepsilon_{\mathrm{hs}} + \varepsilon_{\mathrm{pl}} + \varepsilon_{\mathrm{cr}} + \varepsilon_{\mathrm{vp}}$$

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:

$$\boldsymbol{F} = \boldsymbol{F}_{\mathrm{el}}\boldsymbol{F}_{\mathrm{inel}} \tag{3-4}$$

The order is important here, multiplication from the left makes the elastic deformation act on the inelastically deformed state.

Since a deformation gradient tensor describes a mapping from one frame to another, there are actually three frames involved in this operation. The $\boldsymbol{F}$ tensor is defined by the displacements as usual and describes the mapping from the material frame to the spatial frame. The $\boldsymbol{F}_{\mathrm{inel}}$ tensor, however, describes a mapping from the material frame to an intermediate frame, and the $\boldsymbol{F}_{\mathrm{el}}$ tensor describes a mapping from the intermediate frame to the spatial frame.

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

$$\boldsymbol{F}_{\mathrm{el}} = \boldsymbol{F}\boldsymbol{F}_{\mathrm{inel}}^{-1} \tag{3-5}$$

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

$$\boldsymbol{C}_{\mathrm{el}} = \boldsymbol{F}_{\mathrm{el}}^{T}\boldsymbol{F}_{\mathrm{el}}$$

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

$$\varepsilon_{\mathrm{el}} = \frac{1}{2}(\boldsymbol{C}_{\mathrm{el}} - \boldsymbol{I})$$

The inelastic deformation tensor $\boldsymbol{F}_{\mathrm{inel}}$ is derived from inelastic processes, such as thermal expansion or plasticity. When there are several inelastic contributions, they are applied sequentially to obtain the total inelastic deformation tensor $\boldsymbol{F}_{\mathrm{inel}}$.

$$\boldsymbol{F}_{\mathrm{inel}} = \boldsymbol{F}_1\boldsymbol{F}_2\boldsymbol{F}_3\boldsymbol{F}_4 \ \ldots$$

where $\boldsymbol{F}_i$ is the inelastic strain contribution from subnode i under a Linear Elastic Material, Nonlinear Elastic Material, or Hyperelastic Material.

- The order is important when deformations are finite. The contributions are applied in the same order as the subnodes appear in the model tree. If a **Thermal Expansion** node appears before a **Plasticity** node, then the physical process can be viewed as a thermal expansion followed by a plastic deformation.
- When a certain inelastic strain contribution is small, the order is not significant.
- If the inelastic strain is a pure isotropic volume change, as is often the case for thermal expansion and hygroscopic swelling, the order is not significant.

The internal variables for the elastic right Cauchy-Green deformation tensor in the local coordinate system are named `solid.Cel11`, `solid.Cel12`, and so on; and for the elastic Green-Lagrange tensor in local coordinates `solid.eel11`, `solid.eel12`, and so on.

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

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

The internal variables for the elastic, inelastic, and total volume ratio are named `solid.Jel`, `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, $\boldsymbol{F}_{\mathrm{pl}}$.

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

$$\varepsilon_{\mathrm{pl}} = \frac{1}{2}(\boldsymbol{F}_{\mathrm{pl}}^T\boldsymbol{F}_{\mathrm{pl}} - \boldsymbol{I})$$

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

$$\varepsilon_{\mathrm{el}} = F_{\mathrm{pl}}^{-T}(\varepsilon - \varepsilon_{\mathrm{pl}})F_{\mathrm{pl}}^{-1}$$

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

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

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

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

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

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

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

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

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

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

For any tensor, the physical components are defined as

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

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

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

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

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

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

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

and the Green-Lagrange strain tensor is

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

Under the axial symmetry assumptions, the covariant components of 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 ε 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

$$\varepsilon_{\mathrm{vol}} = I_1(\varepsilon) = \varepsilon_r + \varepsilon_{\varphi} + \varepsilon_z$$

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 wave length L and phase velocity c:

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

Note that in contrast to the prestress, the perturbation amplitude can have non-zero 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 wave length, and thus the wave number k_Z, can be considered as a parameter. Then, ω can be computed by an eigenfrequency analysis for the 2D cross-section with all three displacement components taken as dependent variables. As a result, one obtains

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

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

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

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

which determine the wave length 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 a circumferential 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 non-zero 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,

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

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 the 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

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

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

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

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

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

The pressure is defined as

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

and is thus positive in compression.

- The internal variables for the invariants I_1, I_2, and I_3 are named `solid.I1s`, `solid.I2s`, and `solid.I3s`, respectively.
- The internal variables for the invariants J_2 and J_3 are named `solid.II2s` and `solid.II3s`, respectively.
- The internal variable for the pressure is `solid.pm`.

PRINCIPAL STRESSES

The principal stresses are the eigenvalues of the stress tensor, computed from the eigenvalue equation.

$$(\sigma - \sigma_p I)\mathbf{v}_p = 0$$

The three principal stresses are ordered so that

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

This ordering is true also for the 2D cases. The corresponding principal directions $\mathbf{v}_{pi}$ are orthonormal.

- The internal variables for the principal stresses are named `solid.sp1`, `solid.sp2`, and `solid.sp3` 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

$$\begin{aligned} I_1(\sigma) &= \sigma_{p1} + \sigma_{p2} + \sigma_{p3} \\ I_2(\sigma) &= \sigma_{p1}\sigma_{p2} + \sigma_{p2}\sigma_{p3} + \sigma_{p1}\sigma_{p3} \\ I_3(\sigma) &= \sigma_{p1}\sigma_{p2}\sigma_{p3} \end{aligned}$$

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.

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

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)

$$\xi = I_1/\sqrt{3} \text{ and radius } r = \sqrt{2J_2}.$$

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 $\sigma_{oct} = I_1/3$, and the shear stress on that plane is defined by

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

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 $\sigma_{p1} \ge \sigma_{p2} \ge \sigma_{p3}$, 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

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \tag{3-9}$$

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 area 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

$$s = s_0 + \mathrm{C} : (\varepsilon - \varepsilon_0)$$

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 Material Models, Piezoelectric Material, or Cam-Clay Material Model.

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

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

The **External Stress** subnode can be added to Linear Elastic Material, Nonlinear Elastic Material Models, Piezoelectric Material, or Cam-Clay Material Model.

Axial Symmetry and Stresses

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

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

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

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

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

The first invariant of the stress tensor is

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

Equation of Motion

The first Piola-Kirchhoff stress P is calculated from the second Piola-Kirchhoff stress as $P = FS$. The first Piola-Kirchhoff stress relates forces in the present configuration with areas in the reference configuration, and it is sometimes called the *nominal stress*.

Using the first Piola-Kirchhoff stress tensor, the equation of motion can be written in the following form:

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

where the density corresponds to the material density in the initial undeformed state, the volume force vector $\mathbf{F}_V$ has components in the actual configuration but given with respect to the undeformed volume, and the tensor divergence operator is computed with respect to the coordinates on the material frame. Equation 3-10 is the strong form that corresponds to the weak form equations within the Solid Mechanics interface (and many related multiphysics interfaces) in COMSOL Multiphysics. Using vector and tensor components, the equation can be written as

$$\rho_0 \frac{\partial^2 u_x}{\partial t^2} = F_{Vx} - \left(\frac{\partial P_{xX}}{\partial X} + \frac{\partial P_{xY}}{\partial Y} + \frac{\partial P_{xZ}}{\partial Z}\right)$$

$$\rho_0 \frac{\partial^2 u_y}{\partial t^2} = F_{Vy} - \left(\frac{\partial P_{yX}}{\partial X} + \frac{\partial P_{yY}}{\partial Y} + \frac{\partial P_{yZ}}{\partial Z}\right)$$

$$\rho_0 \frac{\partial^2 u_z}{\partial t^2} = F_{Vz} - \left(\frac{\partial P_{zX}}{\partial X} + \frac{\partial P_{zY}}{\partial Y} + \frac{\partial P_{zZ}}{\partial Z}\right)$$

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 indexes correspond to different frames. Such tensors are called two-point tensors.

The boundary load vector $\mathbf{F}_\mathrm{A}$ 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, s, can be calculated as

$$s = J^{-1}PF^T = J^{-1}FSF^T$$

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

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

where the density corresponds to the density in the actual deformed state, the volume force vector $\mathbf{f}_\mathrm{V}$ 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

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

which corresponds to the volumetric part of the Cauchy stress. The deviatoric part is defined as

$$s_\mathrm{d} = s + p\mathbf{I}$$

The second invariant of the deviatoric stress

$$J_2(s) = \frac{1}{2}s_\mathrm{d}:s_\mathrm{d}$$

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

$$s_{\text{mises}} = \sqrt{3J_2(s)}$$

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 Material Models
  - 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
- 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
- Cam-Clay Material Model
- Creep and Viscoplasticity
  - Norton Law (Power law)
  - Norton-Bailey Law
  - Garofalo Law (Hyperbolic Sine Law)
  - Navarro-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 Material Model
- Safety Factor Evaluation

You can also add a material model which you have coded yourself and made available as a binary library file using an External Stress-Strain Relation.

In the *COMSOL Multiphysics Reference Manual*:

- Working with External Materials
- External Material

Linear Elastic Material

For a linear elastic material, Hooke's law relates the stress tensor to the elastic strain tensor:

$$\sigma = \sigma_{\mathrm{ex}} + \mathsf{C} : \varepsilon_{\mathrm{el}} = \sigma_{\mathrm{ex}} + \mathsf{C} : (\varepsilon - \varepsilon_{\mathrm{inel}}) \qquad (3\text{-}11)$$

where C is the 4th order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction). The elastic strain $\varepsilon_{\mathrm{el}}$ is the difference between the total strain ε and all inelastic strains $\varepsilon_{\mathrm{inel}}$. There may also be an extra stress contribution σ_{ex} with contributions from initial stresses and viscoelastic stresses. In case of geometric nonlinearity, the second Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor are used.

The elastic strain energy density is

$$W_{\mathrm{s}} = \frac{1}{2}\varepsilon_{\mathrm{el}} : (\mathsf{C} : \varepsilon_{\mathrm{el}} + 2\sigma_0) = \frac{1}{2}\varepsilon_{\mathrm{el}} : (\sigma + \sigma_0) \qquad (3\text{-}12)$$

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

TABLE 3-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Lamé parameter λ	$\lambda =$	$\dfrac{E\nu}{(1+\nu)(1-2\nu)}$	$\dfrac{G(E-2G)}{3G-E}$	$K-\dfrac{2G}{3}$	λ
Lamé parameter μ	$\mu =$	$\dfrac{E}{2(1+\nu)}$	G	G	μ
Pressure-wave speed	$c_\mathrm{p} =$	$\sqrt{\dfrac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\dfrac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\dfrac{K+4G/3}{\rho}}$	$\sqrt{\dfrac{\lambda+2\mu}{\rho}}$
Shear-wave speed	$c_\mathrm{s} =$	$\sqrt{\dfrac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

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:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix}_{\mathrm{ex}} + D\left(\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix} - \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix}_{\mathrm{inel}} \right)$$

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:

$$D_{11} = \frac{E_x^2(E_z\nu_{yz}^2 - E_y)}{D_{\mathrm{denom}}}, \qquad D_{12} = -\frac{E_xE_y(E_z\nu_{yz}\nu_{xz} + E_y\nu_{xy})}{D_{\mathrm{denom}}}$$

$$D_{13} = -\frac{E_xE_yE_y(\nu_{xy}\nu_{yz} + \nu_{xz})}{D_{\mathrm{denom}}}, \qquad D_{22} = \frac{E_y^2(E_z\nu_{xz}^2 - E_x)}{D_{\mathrm{denom}}}$$

$$D_{23} = -\frac{E_yE_z(E_y\nu_{xy}\nu_{xz} + E_x\nu_{yz})}{D_{\mathrm{denom}}}, \qquad D_{33} = \frac{E_yE_z(E_y\nu_{xy}^2 - E_x)}{D_{\mathrm{denom}}}$$

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

where

$$D_{\mathrm{denom}} = E_yE_z\nu_{xz}^2 - E_xE_y + 2\nu_{xy}\nu_{yz}\nu_{xz}E_yE_z + E_xE_z\nu_{yz}^2 + E_y^2\nu_{xy}^2$$

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

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

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

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

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

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

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

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

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

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

$$\mathsf{C}_{ijkl}^{\mathrm{phys}}$$

and the corresponding tensor components are computed internally according to:

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

For an isotropic material:

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

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

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

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

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

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

The energy variation is computed as

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

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

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

and the entropy per unit volume can be calculated as

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

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

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

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

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

The heat balance equation can be written as

$$\rho C_\mathrm{p}\frac{\partial T}{\partial t} + T\frac{\partial}{\partial t}S_\mathrm{elast} = \nabla \cdot (k\nabla T) + Q_\mathrm{h}$$

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

$$Q_\mathrm{h} = \tau : \dot{\varepsilon}$$

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:

$$\rho C_{\mathrm{p}}\frac{\partial T}{\partial t} + \sum_{m,n} T\alpha_{mn}\frac{\partial}{\partial t}\sigma_{mn} = \nabla\cdot(k\nabla T) + Q_{\mathrm{h}} \tag{3-13}$$

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.

Nonlinear Elastic Material Models

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:

$$\sigma = \sigma_{\mathrm{ex}} + \mathrm{C} : \varepsilon_{\mathrm{el}} = \sigma_{\mathrm{ex}} + \mathrm{C} : (\varepsilon - \varepsilon_{\mathrm{inel}})$$

For a more detailed discussion, see Equation 3-11.

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

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

and

$$\varepsilon_{\mathrm{el}} = \mathrm{dev}(\varepsilon_{\mathrm{el}}) + \frac{1}{3}\mathrm{trace}(\varepsilon_{\mathrm{el}})I$$

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

$$\sigma = \mathrm{dev}(\sigma) + \frac{1}{3}\mathrm{trace}(\sigma)I = 2G\mathrm{dev}(\varepsilon_{\mathrm{el}}) + K\mathrm{trace}(\varepsilon_{\mathrm{el}})I$$

where K is the bulk modulus and G is the shear modulus. By using the convention that the pressure is the mean stress defined as positive in compression,

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

The volumetric strain (positive in tension) is

$$\varepsilon_{\mathrm{el,vol}} = \mathrm{trace}(\varepsilon_{\mathrm{el}})$$

The linear relation between pressure and volumetric elastic strain is thus

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

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

$$\mathrm{dev}(\sigma) = 2G\mathrm{dev}(\varepsilon_{\mathrm{el}})$$

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

$$J_2(\sigma) = \frac{1}{2}\mathrm{dev}(\sigma):\mathrm{dev}(\sigma) = (2G)^2 J_2(\varepsilon_{\mathrm{el}})$$

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

We can then write

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

and

$$J_2(\varepsilon_{\mathrm{el}}) = \frac{1}{2}\mathrm{dev}(\varepsilon_{\mathrm{el}}):\mathrm{dev}(\varepsilon_{\mathrm{el}}) = \frac{1}{4}\gamma_{\mathrm{el}}^2$$

The shear stress on the plane $\tau = \sqrt{J_2(\sigma)}$ is then related to the elastic shear strain $\gamma_{\mathrm{el}} = 2\varepsilon_{12} = 2\sqrt{J_2(\varepsilon_{\mathrm{el}})}$ by the shear modulus

$$\tau = G\gamma_{\mathrm{el}}$$

Nonlinear Moduli

For nonlinear elastic materials, there is a nonlinear relation between shear stress and shear strain and/or a nonlinear relation between pressure and volumetric strain.

For the purpose of this discussion, $\tau = \sqrt{J_2(\sigma)}$ and $\gamma_{\mathrm{el}} = 2\sqrt{J_2(\varepsilon_{\mathrm{el}})}$ are used alternatively as variables.

In the most general case:

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

Tangent and secant moduli

The tangent shear modulus $G_{\mathrm{t}}(\gamma_{\mathrm{el}})$ and the secant shear modulus $G_{\mathrm{s}}(\gamma_{\mathrm{el}})$ in the most general case depend non-linearly on the shear strain, and are defined as

$$G_{\mathrm{t}} = \frac{\partial\tau}{\partial\gamma_{\mathrm{el}}} \text{ and } G_{\mathrm{s}} = \frac{\tau}{\gamma_{\mathrm{el}}}$$

The tangent bulk modulus $K_{\mathrm{t}}(\varepsilon_{\mathrm{el,vol}})$ and the secant bulk modulus $K_{\mathrm{s}}(\varepsilon_{\mathrm{el,vol}})$ depend on the elastic volumetric strain, and are defined as

$$K_{\mathrm{t}} = -\frac{\partial p}{\partial\varepsilon_{\mathrm{el,vol}}} \text{ and } K_{\mathrm{s}} = -\frac{p}{\varepsilon_{\mathrm{el,vol}}}$$

For linear elastic materials, it is clear that $G_{\mathrm{t}} = G_{\mathrm{s}} = G$ and $K_{\mathrm{t}} = K_{\mathrm{s}} = K$, but this is not the case for nonlinear elastic materials.

At zero strain, the secant and shear moduli are equal to each other $G_{\mathrm{s}}(0) = G_{\mathrm{t}}(0)$ and $K_{\mathrm{s}}(0) = K_{\mathrm{t}}(0)$.

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 non-linear 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

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

Here, E means the initial Young's modulus, and $\varepsilon_{\mathrm{ref}}$ is the strain at a reference stress σ_{ref}. The parameter n is the stress exponent. It is common to use $\varepsilon_{\mathrm{ref}} = 0.002$, 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

$$\varepsilon_{\mathrm{el}} = \frac{\sigma}{E}$$

and the nonlinear strain by

$$\varepsilon_{\mathrm{nl}} = \varepsilon_{\mathrm{ref}}\left(\frac{\sigma}{\sigma_{\mathrm{ref}}}\right)^n$$

The total strain is the sum of linear an nonlinear strains

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

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

Ramberg-Osgood material in soil engineering

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

$$\varepsilon = \frac{\sigma}{E} + \alpha\frac{\sigma_{\mathrm{ref}}}{E}\left(\frac{\sigma}{\sigma_{\mathrm{ref}}}\right)^{n} \tag{3-14}$$

so at the reference stress σ_{ref}, the strain is $\varepsilon = (1+\alpha)\sigma_{\mathrm{ref}}/E$. 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 $\varepsilon = \sigma_{\mathrm{ref}}/E_{0.7}$.

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)

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

The secant shear modulus is given by the power law relation

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

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:

$$K = K_{\mathrm{t}} \text{ for } \varepsilon_{\mathrm{el,vol}} > 0$$

and

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

where $\varepsilon_{\mathrm{el,vol}}$ 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. This material model allows you to assume either a constant Poisson's ratio, or a constant bulk modulus.

For the uniaxial tensile test, the principal stresses are $\sigma_1 = \sigma_{\mathrm{mises}}$, $\sigma_2 = \sigma_3 = 0$. The principal (axial) strain is positive, and the other two strains (transverse) are negative and related by the Poisson's ratio $\varepsilon_2 = \varepsilon_3 = -\nu\varepsilon_1$.

For uniaxial compression, the axial strain is negative, and the other two strains (transverse) are positive.

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

$$\sigma_{\mathrm{ax}} = E_{\mathrm{s}}\varepsilon_{\mathrm{ax}}$$

where E_{s} is the secant Young's modulus, and the axial stress σ_{ax} is considered as a function of axial strain $\varepsilon_{\mathrm{ax}}$. Thus

$$E_{\mathrm{s}} = \frac{\sigma_{\mathrm{ax}}}{\varepsilon_{\mathrm{ax}}}$$

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

$$E_{\mathrm{s}} = \left.\frac{d\sigma_{\mathrm{ax}}}{d\varepsilon_{\mathrm{ax}}}\right|_{\varepsilon_{\mathrm{ax}} = 0}$$

Assuming a constant Poisson's ratio, the secant shear modulus is then defined as

$$G_{\mathrm{s}} = \frac{E_{\mathrm{s}}}{2(1+\nu)}$$

and the secant bulk modulus as

$$K_\mathrm{s} = \frac{E_\mathrm{s}}{3(1-2\nu)}$$

When using the constant bulk modulus assumption, the secant shear modulus is instead defined as

$$G_\mathrm{s} = \frac{3KE_\mathrm{s}}{9K-E_\mathrm{s}}$$

HYPERBOLIC LAW

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

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

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

$$G_\mathrm{s} = G\frac{1}{1+\frac{\gamma}{\gamma_\mathrm{ref}}}$$

This nonlinear soil model is commonly used for modeling soil dynamics in earthquake engineering problems.

Since $\tau = G_\mathrm{s}\gamma$, the shear stress is bounded by $\tau_\mathrm{max} = G\gamma_\mathrm{ref}$ 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

$$\gamma_{\mathrm{ref}} = \frac{7}{3}\gamma_{0,7}$$

and the secant shear modulus as

$$G_{\mathrm{s}} = G\frac{1}{1+\frac{3}{7}\frac{\gamma}{\gamma_{0,7}}}$$

so that when $\gamma = \gamma_{0.7}$ the secant shear modulus is $G_{\mathrm{s}} = 0.7G$.

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

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

here a and b are material parameters obtained by curve fitting data from the triaxial test. The parameter a is related to the initial Young's modulus E

$$a = \frac{1}{E}$$

and the parameter b defines the asymptote of the hyperbola, which is related to the ultimate value of $\sigma_1 - \sigma_3$ denoted q_{ult}

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

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

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

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

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

which is an hyperbolic law with a secant shear modulus of

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

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

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

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

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

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

USER DEFINED

The user defined option allows you to write explicitly how the pressure depends on the elastic volumetric strain. This could be an analytic function or data interpolated from a table.

The elastic volumetric strain $\varepsilon_{\mathrm{el,vol}}$ 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

$$\sigma = C : \varepsilon_{\mathrm{el}}$$

where the elastic strain tensor $\varepsilon_{\mathrm{el}} = \varepsilon - \varepsilon_{\mathrm{inel}}$ 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:

$$\sigma = -pI + \sigma_{\mathrm{d}}$$

The pressure, mean stress, or volumetric stress, is given with a positive sign in compression

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

and the deviatoric stress is computed from the total stress minus the volumetric contribution

$$\sigma_{\mathrm{d}} = \mathrm{dev}(\sigma) = \sigma + pI$$

The elastic strain tensor $\varepsilon_{\mathrm{el}}$ can in the same way be decomposed into volumetric and deviatoric components

$$\varepsilon_{\mathrm{el}} = \frac{1}{3}\varepsilon_{\mathrm{el,vol}}I + \varepsilon_{\mathrm{d}}$$

with the volumetric elastic strain given by

$$\varepsilon_{\mathrm{el,vol}} = \mathrm{trace}(\varepsilon_{\mathrm{el}})$$

and the deviatoric contribution by

$$\varepsilon_{\mathrm{d}} = \mathrm{dev}(\varepsilon_{\mathrm{el}})$$

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

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

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

$$\sigma_{\mathrm{d}} = \mathrm{dev}(\sigma) = 2G\mathrm{dev}(\varepsilon_{\mathrm{el}}) = 2G\varepsilon_{\mathrm{d}}$$

The total stress in Equation 3-11 is then

$$\sigma = K\varepsilon_{\mathrm{el,vol}}I + 2G\varepsilon_{\mathrm{d}}$$

In case of geometric nonlinearity, σ represents the second Piola-Kirchhoff stress tensor and $\varepsilon_{\mathrm{el}}$ 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:

$$\sigma_{\mathrm{d}} = 2\int_0^t \Gamma(t-t')\frac{\partial\varepsilon_{\mathrm{d}}}{\partial t'}dt'$$

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

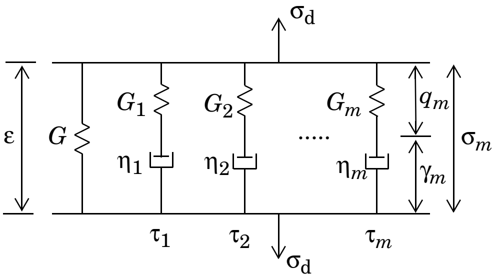


Figure 3-1: Generalized Maxwell model.

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 $\gamma_m = \varepsilon - q_m$ represent the extensions in the dashpots.

The shear modulus of the elastic branch G is normally called the *long-term shear modulus*, or *steady-state stiffness*, and 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 relaxations time τ_m is normally measured in the frequency domain, so the viscosity of the dashpot is not a physical quantity but instead it is derived from stiffness and relaxation time measurements. The viscosity 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

$$\sigma_\mathrm{q} = \sum_{m=1}^{N}\sigma_m = \sum_{m=1}^{N}2G_m(\varepsilon - \gamma_m)$$

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

$$\sigma = \sigma_0 + \mathrm{C} : \varepsilon_\mathrm{el} + \sigma_\mathrm{q} \qquad (3\text{-}15)$$

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

$$\eta_m\dot{\gamma}_m = G_m(\varepsilon - \gamma_m) \qquad (3\text{-}16)$$

The relation between viscosity and relaxation time is

$$\eta_m = G_m\tau_m$$

so that Equation 3-16 can equivalently be written as

$$\tau_m\dot{\gamma}_m + \gamma_m = \varepsilon \qquad (3\text{-}17)$$

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

$$\dot{W}_\mathrm{v} = \sum_{m=1}^{N} \dot{W}_m$$

In order to compute the dissipated energy density, the variable $\dot{W}_\mathrm{v}$ is integrated over time. For frequency domain studies, the dissipation of viscous forces averaged over a time period $2\pi/\omega$ is computed from the shear loss modulus *G”* as

$$W_\mathrm{v} = \omega G'' \tilde{\varepsilon}_\mathrm{d} : \mathrm{conj}(\tilde{\varepsilon}_\mathrm{d})$$

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:

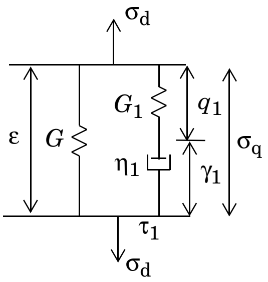


Figure 3-2: Standard linear solid (SLS) model.

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:

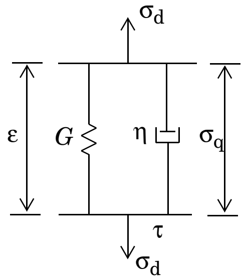


Figure 3-3: Kelvin-Voigt model.

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

$$\sigma_q = 2\eta\dot{\varepsilon} = 2G\tau\dot{\varepsilon} \tag{3-18}$$

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

$$\dot{W}_\mathrm{v} = \sigma_q : \dot{\varepsilon}$$

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 3-17 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

$$\sigma_\mathrm{d} = \mathrm{real}(\tilde{s}_\mathrm{d} e^{j\omega t})$$

$$\varepsilon_\mathrm{d} = \mathrm{real}(\tilde{\varepsilon}_\mathrm{d} e^{j\omega t})$$

Equation 3-15 and Equation 3-17 are then simplified to

$$\tilde{\sigma}_\mathrm{d} = 2(G' + jG'')\tilde{\varepsilon}_\mathrm{d}$$

where the *shear storage modulus* *G’* and the *shear loss modulus* *G”* are defined for the generalized Maxwell model as

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

and for the SLS model as

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

and for the Kelvin-Voigt model as

$$G' = G \text{ and } G'' = \eta\omega = G\tau\omega$$

The internal work of viscous forces averaged over a time period $2\pi/\omega$ is computed as

$$Q_\mathrm{h} = \omega G''\tilde{\varepsilon}_\mathrm{d} : \mathtt{conj}(\tilde{\varepsilon}_\mathrm{d})$$

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

$$W_\mathrm{s} = W_\mathrm{iso} + W_\mathrm{vol} + \sum_{m=1}^{N} \Psi_m$$

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

The second Piola-Kirchhoff stress is computed from

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

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

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

The time evolution of the auxiliary stress tensor $\boldsymbol{Q}_m$ in each viscoelastic branch is given by the rate

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

here, $\boldsymbol{S}_{\mathrm{iso},m}$ is the isochoric second Piola-Kirchhoff stress tensor in the branch m. These tensors are derived from the strain energy density in the main hyperelastic branch and the *energy factors* β_m as

$$\boldsymbol{S}_{\mathrm{iso},\,m} = 2\frac{\partial W_{\mathrm{iso},\,m}}{\partial \boldsymbol{C}} = 2\beta_m\frac{\partial W_{\mathrm{iso}}}{\partial \boldsymbol{C}} = \beta_m\boldsymbol{S}_{\mathrm{iso}}$$

so the time evolution of the auxiliary stress tensor $\boldsymbol{Q}_m$ is given by

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

This equation is not well suited for modeling prestressed bodies. Applying the change of variables

$$\boldsymbol{q}_m = \beta_m\boldsymbol{S}_{\mathrm{iso}} - \boldsymbol{Q}_m$$

the time evolution of the auxiliary stress tensor $\boldsymbol{q}_m$ reads

$$\tau_m\dot{\boldsymbol{q}}_m + \boldsymbol{q}_m = \beta_m\boldsymbol{S}_{\mathrm{iso}}$$

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* $\boldsymbol{C}$ is used to describe the current state of strain (although one could use the left Cauchy-Green tensor $\boldsymbol{B}$, the deformation gradient tensor $\boldsymbol{F}$, and so forth), so the strain energy density is written as $W_{\mathrm{s}}(\boldsymbol{C})$.

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

$$S = 2\frac{\partial W_\mathrm{s}}{\partial C}$$

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.Sl11 = 2*d(solid.Ws,solid.Cl11)`, `solid.Sl12 = d(solid.Ws,solid.Cl12)` 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

Thermal 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

$$J_\mathrm{el} = \frac{\det(F)}{\det(F_\mathrm{th})} = \frac{J}{J_\mathrm{th}}$$

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

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

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

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

When the coefficient of thermal expansion α is anisotropic, the thermal strain is computed from

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

and the anisotropic thermal gradient is defined as

$$F_{\text{th}} = I + \varepsilon_{\text{th}}$$

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

$$\varepsilon_{\text{hs}} = \beta_{\text{h}}(c_{\text{mo}} - c_{\text{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\text{mo,ref}}$ is the strain-free reference concentration. The coefficient of hygroscopic swelling can represent isotropic or anisotropic swelling. The anisotropic hygroscopic gradient is defined as

$$F_{\text{hs}} = I + \varepsilon_{\text{hs}}$$

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,

$$F_{\text{el}} = FF_{\text{pl}}^{-1}$$

Here, the plastic deformation tensor F_{pl} depends on the plastic flow rule, yield function, and plastic potential.

- Multiplicative Decomposition
- Plastic Flow for Large Strains

ISOCHORIC ELASTIC DEFORMATION

For some classes of hyperelastic materials it is convenient to split the strain energy density into *volumetric* (also called *dilatational*) and *isochoric* (also called *distortional* or *volume-preserving*) contributions. The elastic deformation tensor is then multiplicatively decomposed into the volumetric and isochoric components

$$F_{\mathrm{el}} = F_{\mathrm{el,vol}}\overline{F}_{\mathrm{el}}$$

with $F_{\mathrm{el,vol}}$ as the volumetric elastic deformation (a diagonal tensor) and $\overline{F}_{\mathrm{el}}$ 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

$$J_{\mathrm{el}} = \det(F_{\mathrm{el}}) = \det(F_{\mathrm{el,vol}})$$

and the volumetric deformation as

$$F_{\mathrm{el,vol}} = J_{\mathrm{el}}^{1/3}I$$

By using J_{el} it is possible to define the *isochoric-elastic deformation* gradient

$$\overline{F}_{\mathrm{el}} = J_{\mathrm{el}}^{-1/3}F_{\mathrm{el}}$$

the *isochoric-elastic right Cauchy-Green* tensor

$$\overline{C}_{\mathrm{el}} = \overline{F}_{\mathrm{el}}^{T}\overline{F}_{\mathrm{el}} = J_{\mathrm{el}}^{-2/3}C_{\mathrm{el}}$$

and the *isochoric-elastic Green-Lagrange strain* tensor

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

This scaling changes the eigenvalues of the tensor, but not its principal directions, so the original and isochoric tensors remain coaxial to each other.

Some authors call $\overline{F_{\mathrm{el}}}$ and $\overline{C_{\mathrm{el}}}$ the *modified* tensors. Note that

$$\det(\overline{F_{\mathrm{el}}}) = \det(\overline{C_{\mathrm{el}}}) = 1$$

The internal variables for the isochoric-elastic Cauchy-Green deformation tensor in local coordinate system are named `solid.CIel11`, `solid.CIel12`, and so on.

The other two invariants normally used together with J_{el} are the first and second invariant of the isochoric-elastic right Cauchy-Green deformation tensor

$$I_1(\overline{C_{\mathrm{el}}}) \text{ and } I_2(\overline{C_{\mathrm{el}}})$$

In these equations:

$$I_1(\overline{C_{\mathrm{el}}}) = \mathrm{trace}(\overline{C_{\mathrm{el}}}) = J_{\mathrm{el}}^{-2/3} I_1(C_{\mathrm{el}})$$

$$I_2(\overline{C_{\mathrm{el}}}) = \frac{1}{2}(I_1^2(\overline{C_{\mathrm{el}}}) - \mathrm{trace}(\overline{C_{\mathrm{el}}}^2)) = J_{\mathrm{el}}^{-4/3} I_2(C_{\mathrm{el}})$$

Since $I_3(\overline{C_{\mathrm{el}}}) = \det(\overline{C_{\mathrm{el}}}) = 1$, the third invariant is never explicitly used.

The internal variables for the invariants J_{el}, $I_1(\overline{C_{\mathrm{el}}})$, and $I_2(\overline{C_{\mathrm{el}}})$ are named `solid.Jel`, `solid.I1CIel`, and `solid.I2CIel`.

The invariants of the isochoric (modified) elastic Green-Lagrange strain tensor are related to the invariants of the isochoric-elastic right Cauchy-Green deformation tensor

$$I_1(\overline{\varepsilon_{\mathrm{el}}}) = \mathrm{trace}(\overline{\varepsilon_{\mathrm{el}}}) = \frac{1}{2}(I_1(\overline{C_{\mathrm{el}}}) - 3)$$

$$I_2(\overline{\varepsilon_{\mathrm{el}}}) = \frac{1}{2}(I_1^2(\overline{\varepsilon_{\mathrm{el}}}) - \mathrm{trace}(\overline{\varepsilon_{\mathrm{el}}}^2)) = \frac{1}{4}(I_2(\overline{C_{\mathrm{el}}}) - 2I_1(\overline{C_{\mathrm{el}}}) + 3)$$

$$I_3(\overline{\varepsilon_{\mathrm{el}}}) = \det(\overline{\varepsilon_{\mathrm{el}}}) = \frac{1}{8}(I_1(\overline{C_{\mathrm{el}}}) - I_2(\overline{C_{\mathrm{el}}}))\ \mathrm{s}$$

The internal variables for the invariants of the isochoric elastic Green-Lagrange strain tensor are named `solid.I1eIel`, `solid.I2eIel`, and `solid.I3eIel`.

NEARLY INCOMPRESSIBLE HYPERELASTIC MATERIALS

If the **Nearly incompressible material** check box is selected for the Hyperelastic Material node, the total elastic energy function is split into two parts as:

$$W_{\mathrm{s}} = W_{\mathrm{iso}} + W_{\mathrm{vol}}$$

where 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 $\overline{C_{\mathrm{el}}}$ in the local coordinate system.
- Principal invariants of the isochoric elastic right Cauchy-Green tensor $\overline{C_{\mathrm{el}}}$.
- Principal invariants of the isochoric elastic Green-Lagrange strain $\overline{\varepsilon_{\mathrm{el}}}$.

The volumetric strain energy density, W_{vol}, can only be defined as an expression of the elastic volumetric deformation. The most commonly used form is:

$$W_{\mathrm{vol}}(J_{\mathrm{el}}) = \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2 \qquad (3\text{-}19)$$

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

$$p_{\mathrm{m}} = -\frac{\partial W_{\mathrm{vol}}}{\partial J}$$

When the expression in Equation 3-19 is used, the pressure becomes linearly related to the volume change:

$$p_{\mathrm{m}} = -\kappa(J - 1)$$

The second Piola-Kirchhoff stress is then given by

$$S = -p_{\mathrm{m}}JC^{-1} + 2\frac{\partial W_{\mathrm{iso}}}{\partial C} \tag{3-20}$$

and the Cauchy stress tensor by

$$\sigma = J^{-1}FSF^T = -p_{\mathrm{m}}I + 2J^{-1}F\frac{\partial W_{\mathrm{iso}}}{\partial C}F^T$$

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. 11). Finite elements in solid mechanics are said to “lock” when exhibiting an unphysical response to deformation (Ref. 12). 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 3-20 with a corresponding interpolated pressure help variable p_{w}, which adds extra degrees of freedom to the ones defined by the displacement vector $\mathbf{u}$.

The general procedure is the same as when the **Nearly incompressible material** check box is selected for 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 $I_1(C_{\mathrm{el}})$ (Ref. 10).

$$W_{\mathrm{s}} = \frac{1}{2}\mu(I_1 - 3) - \mu\ln(J_{\mathrm{el}}) + \frac{1}{2}\lambda[\ln(J_{\mathrm{el}})]^2$$

Here, λ and μ are the Lamé coefficients.

The nearly incompressible version uses the isochoric invariant $I_1(\overline{C_{\mathrm{el}}})$ and the initial bulk modulus κ

$$W_s = \frac{1}{2}\mu(\overline{I}_1 - 3) + \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2$$

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, $I_1(\varepsilon_{\mathrm{el}})$ and $I_2(\varepsilon_{\mathrm{el}})$

$$W_s = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2$$

Here, λ and μ are the Lamé parameters.

The nearly incompressible version uses the isochoric invariants $I_1(\overline{\varepsilon_{\mathrm{el}}})$ and $I_2(\overline{\varepsilon_{\mathrm{el}}})$, and the initial bulk modulus κ is calculated from the Lamé parameters $\kappa = \lambda + 2\mu/3$

$$W_s = \frac{1}{2}(\lambda + 2\mu)\overline{I}_1^{\,2} - 2\mu\overline{I}_2 + \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2$$

See also the description of the St Venant-Kirchhoff material model in the Solid Mechanics interface documentation.

MOONEY-RIVLIN, TWO PARAMETERS

Only a nearly incompressible version is available, and the elastic strain energy density is written in terms of the two isochoric invariants of the elastic right Cauchy-Green deformation tensors $I_1(\overline{C_{\mathrm{el}}})$ and $I_2(\overline{C_{\mathrm{el}}})$, and the elastic volume ratio J_{el}

$$W_s = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3) + \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2$$

The material parameters C_{10} and C_{01} are related to the Lamé parameter $\mu = 2(C_{10}+C_{01})$.

See also the description of the Mooney-Rivlin, Two Parameters material model in the Solid Mechanics interface documentation.

MOONEY-RIVLIN, FIVE PARAMETERS

Rivlin and Saunders (Ref. 2) proposed a phenomenological model for small deformations in rubber-based materials on a polynomial expansion of the first two invariants of the elastic right Cauchy-Green deformation, so the strain energy density is written as an infinite series

$$W_\mathrm{s} = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} C_{mn}(I_1-3)^m(I_2-3)^n$$

with $C_{00} = 0$. This material model is sometimes also called *polynomial hyperelastic material*.

In the first-order approximation, the material model recovers the Mooney-Rivlin strain energy density

$$W_\mathrm{s} = C_{10}(I_1-3)+C_{01}(I_2-3)$$

while the second-order approximation incorporates second-order terms

$$W_\mathrm{s} = C_{10}(I_1-3)+C_{01}(I_2-3)+C_{20}(I_1-3)^2+C_{02}(I_2-3)^2+C_{11}(I_1-3)(I_2-3)$$

The nearly incompressible version uses the isochoric invariants of the elastic right Cauchy-Green deformation tensors

$$I_1(\overline{C_\mathrm{el}})\ \text{ and }\ I_2(\overline{C_\mathrm{el}})$$

and it adds a contribution due to the elastic volume ratio. The strain energy density is then computed from

$$W_\mathrm{s} = \sum_{m=0}^{2}\sum_{n=0}^{2} C_{mn}(\overline{I_1}-3)^m(\overline{I_2}-3)^n+\frac{1}{2}\kappa(J_\mathrm{el}-1)^2$$

Here, κ is the initial bulk modulus.

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

$$W_\mathrm{s} = \sum_{m=0}^{3}\sum_{n=0}^{3} C_{mn}(\bar{I}_1-3)^m(\bar{I}_2-3)^n+\frac{1}{2}\kappa(J_\mathrm{el}-1)^2$$

See also the description of the Mooney-Rivlin, Nine Parameters material model in the Solid Mechanics interface documentation.

YEOH

Yeoh proposed (Ref. 1) a phenomenological model in order to fit experimental data of filled rubbers, where Mooney-Rivlin and Neo-Hookean models were to simple to describe the stiffening effect in the large strain regime. The strain energy was fitted to experimental data by means of three parameters, and the first invariant of the elastic right Cauchy-Green deformation tensors $I_1(C_\mathrm{el})$

$$W_\mathrm{s} = c_1(I_1-3)+c_2(I_1-3)^2+c_3(I_1-3)^3$$

The shear modulus depends on the deformation, and it is calculated as

$$\mu = 2\left(\frac{\partial W_\mathrm{s}}{\partial I_1}+\frac{\partial W_\mathrm{s}}{\partial I_2}\right) = 2c_1+4c_2(I_1-3)+6c_3(I_1-3)^2$$

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 $I_1(\overline{C}_\mathrm{el})$, and it adds a contribution from the elastic volume ratio

$$W_s = c_1(\overline{I}_1-3)+c_2(\overline{I}_1-3)^2+c_3(\overline{I}_1-3)^3+\frac{1}{2}\kappa(J_{el}-1)^2$$

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

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

and the initial bulk modulus κ

$$W_s = \sum_{p=1}^{N}\frac{\mu_p}{\alpha_p}(\overline{\lambda}_{el1}^{\alpha_p}+\overline{\lambda}_{el2}^{\alpha_p}+\overline{\lambda}_{el3}^{\alpha_p}-3)+\frac{1}{2}\kappa(J_{el}-1)^2$$

The isochoric elastic stretches define a volume preserving deformation, since

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

The initial shear modulus is

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

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. 12 and Ref. 15) is used to model highly compressible foams. The strain energy density is written in a similar fashion as in Ogden material:

$$W_s = \sum_{k=1}^{N} \frac{2\mu_k}{\alpha_k^2}\left(\lambda_{\mathrm{el}1}^{\alpha_k} + \lambda_{\mathrm{el}2}^{\alpha_k} + \lambda_{\mathrm{el}3}^{\alpha_k} - 3 + \frac{1}{\beta_k}(J_{\mathrm{el}}^{-\alpha_k\beta_k} - 1)\right)$$

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

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

for constant parameters $\beta_k = \beta$, the initial bulk modulus becomes $\kappa = 2\mu(\beta + 1/3)$, so a stable material requires $\mu > 0$ and $\beta > -1/3$. In this case, the Poisson's ratio is given by $\nu = \beta/(2\beta{+}2/3)$, which means that for a Poisson’s ratio larger than −1, $\beta > -2/9$ is needed.

See also the description of the Storakers material model in the Solid Mechanics interface documentation.

VARGA

The Varga material model (Ref. 1) describes the strain energy in terms of the elastic stretches as

$$W_s = c_1(\lambda_{el1} + \lambda_{el2} + \lambda_{el3} - 3) + c_2(\lambda_{el1}\lambda_{el2} + \lambda_{el2}\lambda_{el3} + \lambda_{el1}\lambda_{el3} - 3)$$

The nearly incompressible version uses the isochoric elastic stretches defined as

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

and the initial bulk modulus κ

$$W_s = c_1(\bar{\lambda}_{el1} + \bar{\lambda}_{el2} + \bar{\lambda}_{el3} - 3) + c_2(\bar{\lambda}_{el1}\bar{\lambda}_{el2} + \bar{\lambda}_{el2}\bar{\lambda}_{el3} + \bar{\lambda}_{el1}\bar{\lambda}_{el3} - 3) + \frac{1}{2}\kappa(J_{el} - 1)^2$$

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

$$W_s = \mu(\bar{\lambda}_{el1} + \bar{\lambda}_{el2} + \bar{\lambda}_{el3} - 3) + \frac{1}{2}\kappa(J_{el} - 1)^2$$

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, $I_1(C_{el})$ 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 as 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

The strain energy density is then defined as

$$W_{\mathrm{s}} = \mu_0 \sum_{p=1}^{5} c_p (I_1^p - 3^p)$$

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 $I_1(\overline{C_{\mathrm{el}}})$ and the initial bulk modulus κ

$$W_{\mathrm{s}} = \mu_0 \sum_{p=1}^{5} c_p (I_1^p - 3^p) + \frac{1}{2}\kappa (J_{\mathrm{el}} - 1)^2$$

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. 13 and Ref. 14) is a simple phenomenological constitutive model based on only two parameters, μ and j_{m}, which defines the strain energy density as:

$$W_{\mathrm{s}} = -\frac{\mu}{2} j_{\mathrm{m}} \log\left(1 - \frac{\overline{I_1} - 3}{j_{\mathrm{m}}}\right)$$

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, the 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 $j_{\mathrm{m}} \to \infty$.

The nearly incompressible formulation uses the isochoric invariants $I_1(\overline{C_{\mathrm{el}}})$ and the initial bulk modulus κ:

$$W_{\mathrm{s}} = -\frac{\mu}{2}j_{\mathrm{m}}\log\left(1-\frac{\bar{I}_1-3}{j_{\mathrm{m}}}\right)+\frac{1}{2}\kappa(J_{\mathrm{el}}-1)^2$$

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, $I_1(C_{\mathrm{el}})$, $I_2(C_{\mathrm{el}})$, and $I_3(C_{\mathrm{el}})$

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

Here, ϕ is an interpolation parameter bounded to $0 < \phi < 1$, μ is the shear modulus, and β is an expression of Poisson's ratio.

When the parameter $\beta \to \infty$, the strain energy simplifies to a similar form of the Mooney-Rivlin material model

$$W_{\mathrm{s}} = \phi\frac{\mu}{2}(I_1-3)+(1-\phi)\frac{\mu}{2}\left(\frac{I_2}{I_3}-3\right)$$

In the special case of $\phi = 1$, the strain energy reduces to a similar form of the Neo-Hookean model

$$W_{\mathrm{s}} = \frac{\mu}{2}(I_1-3)+\frac{\mu}{2\beta}(J^{-2\beta}-1)$$

See also the description of the Blatz-Ko material model in the Solid Mechanics interface documentation.

GAO

Gao proposed (Ref. 16) 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}:

$$W_{\mathrm{s}} = a(I_1^n + I_{-1}^n)$$

Here, the invariant $I_{\text{-}1}(C_{\mathrm{el}})$ is calculated as:

$$I_{-1} = \mathrm{trace}(C_{\mathrm{el}}^{-1}) = \frac{I_2(C_{\mathrm{el}})}{I_3(C_{\mathrm{el}})}$$

Gao proposed that the material is unconditionally stable when the parameters are bounded to $3 > n > 1$ and $a > 0$, and related these parameters under small strain to the Young's modulus and Poisson's ratio by:

$$E = \frac{3^n n^2 8a}{2n+1} \text{ and } \nu = \frac{n-1}{2n+1}$$

Since $n = (1+\nu)/(1-2\nu)$ and it is bounded to $1 < n < 3$, this material 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, $I_1(\varepsilon_{\mathrm{el}})$, $I_2(\varepsilon_{\mathrm{el}})$, and $I_3(\varepsilon_{\mathrm{el}})$

$$W_{\mathrm{s}} = \frac{1}{2}(\lambda + 2\mu)I_1^2 - 2\mu I_2 + \frac{1}{3}(l + 2m)I_1^3 - 2mI_1I_2 + nI_3$$

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

When a material model is user-defined, an expression for the elastic strain energy W_s is entered, 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}

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

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

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

The internal variables for these invariants are named `solid.I1Cel`, `solid.I2Cel`, and `solid.I3Cel`.

- Components of the elastic Green-Lagrange strain tensor ε_{el} in the local coordinate system.
- Principal elastic stretches λ_{el1}, λ_{el2}, and λ_{el3}, which are the square-root of the eigenvalues of the elastic right Cauchy-Green deformation tensor C_{el}.

The internal variables for the principal elastic stretches are named `solid.stchelp1`, `solid.stchelp2`, and `solid.stchelp3`.

- Invariants of the elastic Green-Lagrange strain tensor. Since

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

the invariants of ε_{el} are written in terms of the invariants of C_{el}:

$$I_1(\varepsilon_{el}) = \mathrm{trace}(\varepsilon_{el}) = \frac{1}{2}(I_1(C_{el}) - 3)$$

$$I_2(\varepsilon_{el}) = \frac{1}{2}(I_1^2(\varepsilon_{el}) - \mathrm{trace}(\varepsilon_{el}^2)) = \frac{1}{4}(I_2(C_{el}) - 2I_1(C_{el}) + 3)$$

$$I_3(\varepsilon_{\mathrm{el}}) = \det(\varepsilon_{\mathrm{el}}) = \frac{1}{8}(I_3(C_{\mathrm{el}}) - I_2(C_{\mathrm{el}}) + I_1(C_{\mathrm{el}}) - 1)$$

The internal variables for these invariants are named `solid.I1eel`, `solid.I2eel`, and `solid.I3eel`.

- When the **Nearly incompressible material** check box 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

  $$J_{\mathrm{el}} = \det(F_{\mathrm{el}})$$

  The isochoric strain energy, W_{iso}, as an expression involving the invariants of the isochoric elastic right Cauchy-Green tensor

  $$I_1(\overline{C_{\mathrm{el}}}) \text{ and } I_2(\overline{C_{\mathrm{el}}})$$

  or the invariants of the isochoric elastic Green-Lagrange strain

  $$I_1(\overline{\varepsilon_{\mathrm{el}}}), I_2(\overline{\varepsilon_{\mathrm{el}}}), \text{ and } I_3(\overline{\varepsilon_{\mathrm{el}}})$$

The internal variables for J_{el}, $I_1(\overline{C_{\mathrm{el}}})$, and $I_2(\overline{C_{\mathrm{el}}})$ are named `solid.Jel`, `solid.I1CIel`, and `solid.I2CIel`.

The internal variables for $I_1(\overline{\varepsilon_{\mathrm{el}}})$, $I_2(\overline{\varepsilon_{\mathrm{el}}})$, and $I_3(\overline{\varepsilon_{\mathrm{el}}})$ are named `solid.I1eIel`, `solid.I2eIel`, and `solid.I3eIel`.

The strain energy density must not contain any other expressions involving displacement or their derivatives. This excludes components of the displacement gradient $\nabla \mathbf{u}$ and deformation gradient $F = \nabla \mathbf{u} + \mathbf{I}$ tensors, their transpose, inversions, as well as the global material system components of C and ε. If they occur, such variables are treated as constants during symbolic differentiations.

See also the description of the User defined material model in the Solid Mechanics interface documentation.

Elastoplastic Material Models

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 Material Models, 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

$$F = f(\sigma) - f_{\mathrm{c}} = 0$$

where f_c can be a constant value (for perfectly plastic materials), or a variable for strain-hardening materials. The yield surface $\boldsymbol{F}$ is a surface in the space of principal stresses, in which the elastic regime ($\boldsymbol{F} \le 0$) 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 $\boldsymbol{F} = 0$, which can be chosen for numerical purposes as $\boldsymbol{F} = f(\sigma)^2 - f_c^2 = 0$.

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

$$\begin{aligned} I_1(\sigma) &= \mathrm{trace}(\sigma) \\ J_2(\sigma) &= \frac{1}{2}\mathrm{dev}(\sigma):\mathrm{dev}(\sigma) \\ J_3(\sigma) &= \det(\mathrm{dev}(\sigma)) \end{aligned}$$

so that the increment of the plastic strain tensor $\dot{\varepsilon}_p$ can be decomposed into

$$\dot{\varepsilon}_p = \lambda\frac{\partial Q_p}{\partial\sigma} = \lambda\left(\frac{\partial Q_p}{\partial I_1}\frac{\partial I_1}{\partial\sigma} + \frac{\partial Q_p}{\partial J_2}\frac{\partial J_2}{\partial\sigma} + \frac{\partial Q_p}{\partial J_3}\frac{\partial J_3}{\partial\sigma}\right)$$

The increment in the plastic strain tensor $\dot{\varepsilon}_p$ includes in a general case both deviatoric and volumetric parts. The tensor $\dot{\varepsilon}_p$ is symmetric given the following properties

$$\begin{aligned} \frac{\partial I_1}{\partial\sigma} &= I \\ \frac{\partial J_2}{\partial\sigma} &= \mathrm{dev}(\sigma) \\ \frac{\partial J_3}{\partial\sigma} &= \mathrm{dev}(\sigma)\mathrm{dev}(\sigma) - \frac{2}{3}J_2 I \end{aligned} \tag{3-21}$$

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

$$\dot{\varepsilon}_{\mathrm{pe}} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{\mathrm{p}}:\dot{\varepsilon}_{\mathrm{p}}} \tag{3-22}$$

The trace of the incremental plastic strain tensor, which is called the *volumetric plastic strain rate* $\dot{\varepsilon}_{\mathrm{pvol}}$, 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

$$\dot{\varepsilon}_{\mathrm{pvol}} = \mathrm{trace}(\dot{\varepsilon}_{\mathrm{p}}) = \lambda\ \mathrm{trace}\left(\frac{\partial \boldsymbol{Q}_{\mathrm{p}}}{\partial\sigma}\right) = 3\lambda\frac{\partial \boldsymbol{Q}_{\mathrm{p}}}{\partial I_1}$$

For metal plasticity under the von Mises or Tresca criteria, the volumetric plastic strain rate 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 it not the case for most materials used in geotechnical applications. For instance, a nonzero volumetric plastic strain is explicitly used in the Cam-Clay material.

- 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:

$$\boldsymbol{F}_{\mathrm{y}} = \phi(\sigma) - \sigma_{\mathrm{ys}}$$

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:

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

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 by 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 $\sigma_1 \ge \sigma_2 \ge \sigma_3$, 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 $0 \le \theta \le \pi/3$, 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 non-associated 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.

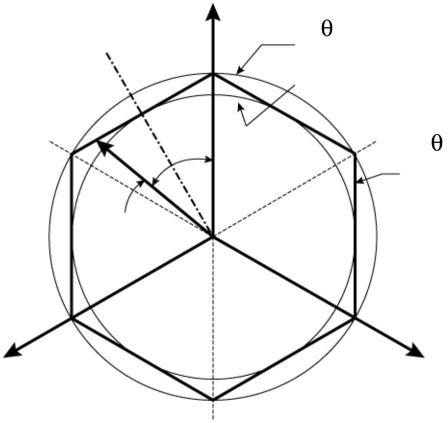


Figure 3-4: The upper and lower limits of the Tresca criterion.

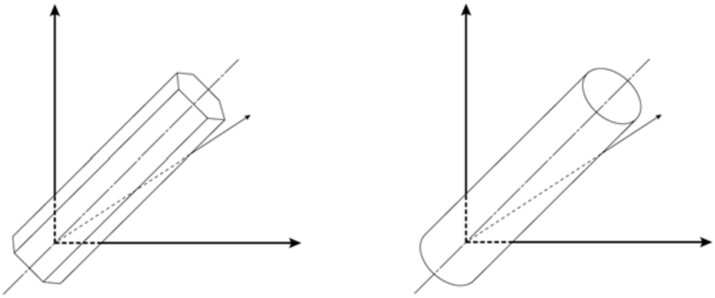


Figure 3-5: Classical yield criteria for metals. Tresca criterion (left) and von Mises criterion (right).

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 *J2* plasticity, as the plastic flow is independent on the mean pressure.

POROUS PLASTICITY

The description of plastic deformation in soils, porous metals and aggregates has a main difference with respect to traditional metal plasticity; as the yield function and plastic potential are not only defined in terms of the deviatoric stress tensor component (or the deviatoric stress invariant J_2), but they also include dependencies on the hydrostatic pressure.

Shima-Oyane Criterion

Shima and Oyane (Ref. 14) proposed a plastic yield surface for modeling the compaction of porous metallic structures fabricated by sintering. This criterion can been 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

$$Q_\mathrm{p}(\sigma) = \left(\frac{\sigma_\mathrm{e}}{\sigma_0}\right)^2 + \alpha(1-\rho_\mathrm{rel})^\gamma\left(\frac{p_\mathrm{m}}{\sigma_0}\right)^2 - \rho_\mathrm{rel}^m$$

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,* which is related to the porosity ϕ by $\rho_\mathrm{rel} = 1 - \phi$. The material parameters α, γ, and m are obtained from curve fitting experimental data. Typical material parameter values for copper powder 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

$$Q_\mathrm{p}(\sigma) = \left(\frac{\sigma_\mathrm{e}}{\sigma_0}\right)^2 + 2\phi\cosh\left(\frac{3p_\mathrm{m}}{2\sigma_0}\right) - (1+\phi^2)$$

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

$$Q_{\mathrm{p}}(\sigma) = \left(\frac{\sigma_{\mathrm{e}}}{\sigma_0}\right)^2 + 2q_1\phi_{\mathrm{e}}\cosh\left(\frac{3q_2p_{\mathrm{m}}}{2\sigma_0}\right) - (1 + q_3\phi_{\mathrm{e}}{}^2)$$

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:

$$\phi_{\mathrm{e}} = \begin{cases} \phi & \text{for} \quad \phi < \phi_{\mathrm{c}} \\ \phi_{\mathrm{c}} + \dfrac{\phi_{\mathrm{m}} - \phi_{\mathrm{c}}}{\phi_{\mathrm{f}} - \phi_{\mathrm{c}}}(\phi - \phi_{\mathrm{c}}) & \text{for } \phi_{\mathrm{c}} < \phi < \phi_{\mathrm{f}} \end{cases}$$

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 value of failure, the effective porosity takes a maximum value of ϕ_{m}; at this high porosity, the porous material loses the capacity to carry stresses. This maximum porosity value is derived from other parameters

$$\phi_{\mathrm{m}} = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3}$$

since typically the correction parameters are related as $q_3 = q_1{}^2$, then $\phi_{\mathrm{m}} = 1/q_1$.

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

$$Q_{\mathrm{p}} = \left(\frac{5}{18}\frac{\sigma_{\mathrm{e}}}{p_{\mathrm{f}}} + \frac{2}{3}\right)^2 + \left(\frac{5}{3}\frac{p_{\mathrm{m}}}{p_{\mathrm{f}}}\right)^2 - 1$$

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

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

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). For low void volume fractions (porosity lower than 10%), the GTN model is used 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.

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

$$\frac{1}{2}(\sigma_1-\sigma_3)+\frac{1}{2}(\sigma_1+\sigma_3)\sin\phi - \mathrm{c}\cos\phi = 0$$

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.

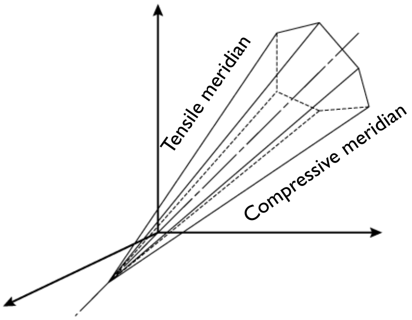


Figure 3-6: The Mohr-Coulomb criterion. The cone opens toward the compressive axis.

The Mohr-Coulomb criterion can be written in terms of the invariants I_1 and J_2 and the Lode angle $0 \le \theta \le \pi/3$ (Ref. 1, Ref. 9) when the principal stresses are sorted as $\sigma_1 \ge \sigma_2 \ge \sigma_3$. The yield function then reads

$$F_\mathrm{y} = \frac{1}{3}I_1\sin\phi + \sqrt{\frac{J_2}{3}}\left((1+\sin\phi)\cos\theta - (1-\sin\phi)\cos\left(\theta + \frac{2\pi}{3}\right)\right) - \mathrm{c}\cos\phi = 0$$

The tensile meridian is defined when $\theta = 0$ and the compressive meridian when $\theta = \pi/3$.

Rearranging terms, the Mohr-Coulomb criterion reads

$$F_\mathrm{y} = \sqrt{J_2}m(\theta) + \alpha I_1 - k = 0$$

where

$$m(\theta) = \cos(\theta - \pi/6) - \sqrt{1/3}\sin\phi\sin(\theta - \pi/6),\ \alpha = \sin\phi/3,\ \text{and}\ k = \mathrm{c}\cos\phi$$

In the special case of frictionless material, ($\phi = 0$, $\alpha = 0$, $k = \mathrm{c}$), 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 $\alpha = 0$.

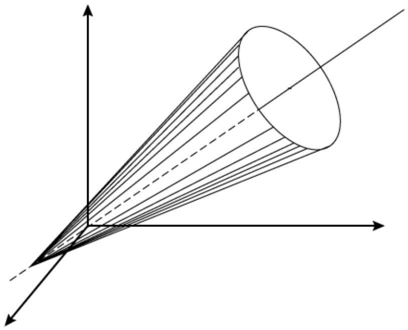


Figure 3-7: The Drucker-Prager criterion. The cone opens toward the compressive axis.

The coefficients in the Drucker-Prager model can be matched to the coefficients in the Mohr-Coulomb criterion by

$$\alpha = \frac{2}{\sqrt{3}} \cdot \frac{\sin\phi}{(3 \pm \sin\phi)} \text{ and } k = \frac{2\sqrt{3}c\cos\phi}{(3 \pm \sin\phi)}$$

The symbol ± 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

$$m(\pi/3) = (3 - \sin\phi)/(2\sqrt{3})$$

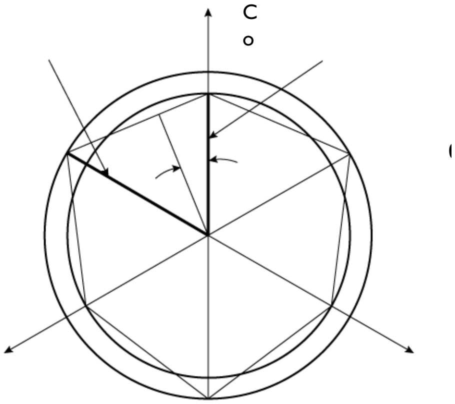


Figure 3-8: The Drucker-Prager criterion showing the tensile and compressive meridians (inner and outer circles), and the Lode angle compared to the cross section of Mohr-Coulomb criterion in the π-plane.

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 matching Drucker-Prager criterion to Mohr-Coulomb criterion in 2D plane-strain applications, the parameters are

$$\alpha = \frac{\tan\phi}{\sqrt{9 + 12\tan^2\phi}} \quad \text{and} \quad k = \frac{3c}{\sqrt{9 + 12\tan^2\phi}}$$

and when matching both criteria in 2D plane-stress applications, 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 non-associated 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 non-associated, in which case the difference between the dilatation angle and the angle of internal friction would result in a yield surface and plastic potential portrayed by two cones with different angles with respect to the hydrostatic angle.

Elliptic Cap

The Mohr-Coulomb and Drucker-Prager criteria portray a conic yield surface which opens in the hydrostatic axis direction. Normally, these soil models are not accurate above a given limit pressure because real-life materials cannot bear infinite loads and still behave elastically. A simple way to overcome this problem is to add an elliptical end-cap on the compressive side to these models.

The elliptic cap is an elliptic yield surface of semi-axes as shown in Figure 3-9. 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

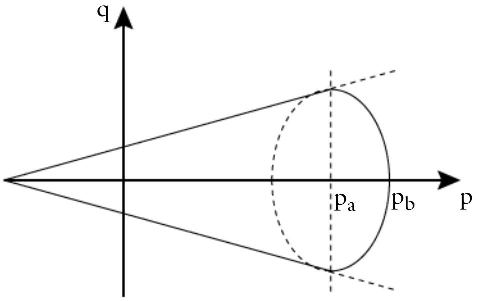


Figure 3-9: Elliptic cap model in Haigh–Westergaard coordinate system.

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 3-9 shows the cap in terms of the variables p and q

$$q = \sqrt{3J_2(\sigma)} \text{ and } p = -I_1(\sigma)/3$$

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 $\varepsilon_{pvol,max}$ 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

$$F_y = (9+9\mu^2)I_3 - I_1 I_2 = 0$$

where the parameter $\mu = (\tau/\sigma_n)_{STP}$ equals the maximum ratio between shear stress and normal stress in the *spatially mobilized plane* (STP-plane), 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.

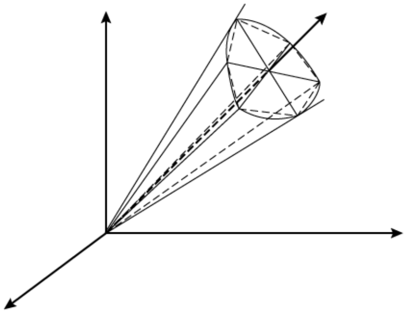


Figure 3-10: The Matsuoka-Nakai criterion and Mohr-Coulomb criterion in the principal stress space.

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

$$F_{\mathrm{y}} = kI_3 - I_1^3 = 0$$

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

$$F_{\mathrm{y}} = J_3 - \frac{1}{3}I_1J_2 + \left(\frac{1}{27} - \frac{1}{k}\right)I_1^3 = 0$$

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

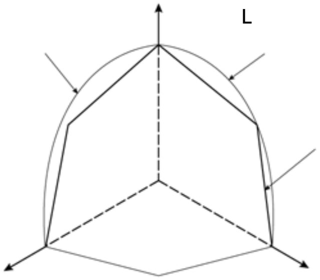


Figure 3-11: Comparing the Mohr-Coulomb, Matsuoka-Nakai, and Lade-Duncan criteria when matching the tensile meridian.

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

$$\sigma_{\mathrm{t}} < c\frac{\cos\phi}{\sin\phi}$$

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 $\mathbf{a}_i$

$$\begin{gathered} Q_{\mathrm{p}} = F(\sigma_{22}-\sigma_{33})^2 + G(\sigma_{33}-\sigma_{11})^2 + H(\sigma_{11}-\sigma_{22})^2 + \\ 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 - 1 \end{gathered} \tag{3-23}$$

The six parameters *F, G, H, L, M*, and *N* are related to the state of anisotropy. As with isotropic plasticity, the elastic region $Q_{\mathrm{p}} < 0$ is bounded by the yield surface $Q_{\mathrm{p}} = 0$.

Hill demonstrated that this type of anisotropic plasticity is volume preserving, this is, given the associated flow rule

$$\dot{\varepsilon}_{\mathrm{p}} = \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma}$$

the trace of the plastic strain rate tensor is zero, which follows from the expressions for the diagonal elements of $\dot{\varepsilon}_{\mathrm{p}}$

$$\begin{aligned} \dot{\varepsilon}_{\mathrm{p}11} &= \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma_{11}} = 2\lambda(-G(\sigma_{33}-\sigma_{11}) + H(\sigma_{11}-\sigma_{22})) \\ \dot{\varepsilon}_{\mathrm{p}22} &= \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma_{22}} = 2\lambda(F(\sigma_{22}-\sigma_{33}) - H(\sigma_{11}-\sigma_{22})) \\ \dot{\varepsilon}_{\mathrm{p}33} &= \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma_{33}} = 2\lambda(-F(\sigma_{22}-\sigma_{33}) + G(\sigma_{33}-\sigma_{11})) \end{aligned}$$

so the plastic volumetric strain rate is zero

$$\dot{\varepsilon}_{\mathrm{pvol}} = \mathrm{trace}(\dot{\varepsilon}_{\mathrm{p}}) = \dot{\varepsilon}_{\mathrm{p11}} + \dot{\varepsilon}_{\mathrm{p22}} + \dot{\varepsilon}_{\mathrm{p33}} = 0$$

Hill plasticity is an extension of *J2* (von Mises) plasticity, in the sense that it is volume preserving. Due to this assumption, six parameters are needed to define orthotropic plasticity, as opposed to orthotropic elasticity, where nine elastic coefficients are needed.

Expressions for the Coefficients F, G, H, L, M, N

Hill noticed that the parameters L, M, and N are related to the yield stress in shear with respect to the axes of orthotropy $\mathbf{a}_i$, thus they are positive parameters

$$L = \frac{1}{2\sigma_{\mathrm{ys23}}^2},\ M = \frac{1}{2\sigma_{\mathrm{ys31}}^2},\ N = \frac{1}{2\sigma_{\mathrm{ys12}}^2}$$

Here, σ_{ysij} represents the yield stress in shear on the plane ij.

The material parameters σ_{ys1}, σ_{ys2}, and σ_{ys3} represent the tensile yield stress in the direction, $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, and they are related to Hill's parameters F, G, and H as

$$\frac{1}{\sigma_{\mathrm{ys1}}^2} = G + H$$
$$\frac{1}{\sigma_{\mathrm{ys2}}^2} = H + F$$
$$\frac{1}{\sigma_{\mathrm{ys3}}^2} = F + G$$

or equivalently

$$2F = \frac{1}{\sigma_{\mathrm{ys2}}^2} + \frac{1}{\sigma_{\mathrm{ys3}}^2} - \frac{1}{\sigma_{\mathrm{ys1}}^2}$$
$$2G = \frac{1}{\sigma_{\mathrm{ys3}}^2} + \frac{1}{\sigma_{\mathrm{ys1}}^2} - \frac{1}{\sigma_{\mathrm{ys2}}^2}$$
$$2H = \frac{1}{\sigma_{\mathrm{ys1}}^2} + \frac{1}{\sigma_{\mathrm{ys2}}^2} - \frac{1}{\sigma_{\mathrm{ys3}}^2}$$

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)

$$\frac{1}{\sigma_{ys0}^2} = \frac{2}{3}(F+G+H) = \frac{1}{3}\left(\frac{1}{\sigma_{\mathrm{ys1}}^2}+\frac{1}{\sigma_{\mathrm{ys2}}^2}+\frac{1}{\sigma_{\mathrm{ys3}}^2}\right) \tag{3-24}$$

Defining *Hill's effective stress* as (Ref. 13)

$$\sigma_{\mathrm{hill}}^2 = \sigma_{\mathrm{ys0}}^2(F(\sigma_{22}-\sigma_{33})^2+G(\sigma_{33}-\sigma_{11})^2+H(\sigma_{11}-\sigma_{22})^2 \\ +2(L\sigma_{23}^2+M\sigma_{31}^2+N\sigma_{12}^2))$$

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

$$Q_{\mathrm{p}} = \sigma_{\mathrm{hill}} - \sigma_{\mathrm{ys}}$$

Here, the average yield stress

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$$

now depends on the initial yield stress σ_{ys0}, the hardening function σ_{h}, and the effective plastic strain $\varepsilon_{\mathrm{pe}}$.

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 (or Ideal) Plasticity

For perfect 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 non-associated flow rule for the plastic potential $\boldsymbol{Q}_\mathrm{p}$

$$\dot{\varepsilon}_\mathrm{p} = \lambda\frac{\partial \boldsymbol{Q}_\mathrm{p}}{\partial\sigma}$$

with the yield function

$$\boldsymbol{F}_\mathrm{y} = \phi(\sigma) - \sigma_{\mathrm{ys0}}$$

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 non-associated flow rule is applied as written in Equation 3-30.

Linear Isotropic Hardening

In this case the plasticity algorithm solves either the associated or non-associated flow rule for the plastic potential $\boldsymbol{Q}_\mathrm{p}$

$$\dot{\varepsilon}_\mathrm{p} = \lambda\frac{\partial \boldsymbol{Q}_\mathrm{p}}{\partial\sigma}$$

with the yield function

$$\boldsymbol{F}_\mathrm{y} = \phi(\sigma) - \sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}})$$

where the yield stress $\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}})$ now depends on the *effective plastic strain* $\varepsilon_{\mathrm{pe}}$.

The yield stress $\sigma_{ys}(\varepsilon_{pe})$ is then a function of the effective plastic strain and the *initial yield stress* σ_{ys0}

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + E_{iso}\varepsilon_{pe}$$

here, the *isotropic hardening modulus* E_{iso} is 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 $\sigma_{ys}(\varepsilon_{pe})$ 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

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + k{\varepsilon_{pe}}^{n}$$

here, k is the strength coefficient and n is the hardening exponent. Setting $n = 1$ would result in linear isotropic hardening.

Swift

For non-saturating 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

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_h(\varepsilon_{pe}) = k(\varepsilon_0 + \varepsilon_{pe})^{n}$$

here, k is the strength coefficient, n is the hardening exponent and ε_0 is a reference strain. Noting that at zero plastic strain the initial yield stress is related to the strength coefficient and hardening exponent as

$$\sigma_{ys}(0) = \sigma_{ys0} = k{\varepsilon_0}^{n}$$

the yield stress is written as

$$\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}}\left(1 + \frac{\varepsilon_{\mathrm{pe}}}{\varepsilon_0}\right)^{\mathrm{n}}$$

Voce

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

$$\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{sat}}\left(1 - e^{-\beta\varepsilon_{\mathrm{pe}}}\right)$$

The yield stress $\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}})$ is then defined as

$$\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{sat}}\left(1 - e^{-\beta\varepsilon_{\mathrm{pe}}}\right)$$

The value of the *saturation exponent* parameter β determines the saturation rate of the hysteresis loop for cyclic loading. The *saturation flow stress* σ_{sat} characterizes the maximum distance by which the yield surface can expand in the stress space. For values $\varepsilon_{\mathrm{pe}} >> 1/\beta$, the yield stress saturates to

$$\sigma_{\mathrm{ys}} \to \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{sat}}$$

Hockett-Sherby

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

$$\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}} + (\sigma_{\infty} - \sigma_{\mathrm{ys0}})\left(1 - e^{-\mathrm{m}\varepsilon^{n}_{\mathrm{pe}}}\right)$$

where σ_{∞} is the *steady-state flow stress*, m the saturation coefficient and n the saturation exponent. For values $\mathrm{m}\varepsilon_{\mathrm{pe}}{}^{\mathrm{n}} >> 1$ the yield stress saturates to

$$\sigma_{\mathrm{ys}} \to \sigma_{\infty}$$

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 $\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$ such that the yields stress reads

$$\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$$

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 non-associated flow rule is applied as written in Equation 3-30.

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 non-associated flow rule for the plastic potential Q_{p}

$$\dot{\varepsilon}_{\mathrm{p}} = \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma}$$

with the yield function defined as

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

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:

$$\sigma_{\mathrm{back}} = \frac{2}{3}C_{\mathrm{k}}\varepsilon_{\mathrm{p}}$$

where the ***kinematic hardening modulus*** C_{k} is calculated from

$$\frac{1}{C_{\mathrm{k}}} = \frac{1}{E_{\mathrm{k}}} - \frac{1}{E}$$

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 $H_{\mathrm{k}} = 2/3C_{\mathrm{k}}$.

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, $\sigma_{\mathrm{eff}} = \sigma - \sigma_{\mathrm{back}}$. 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

$$\dot{\sigma}_{\mathrm{back}} = \frac{2}{3}C_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{p}} - \gamma_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{pe}}\sigma_{\mathrm{back}}$$

here, C_{k} is the kinematic hardening modulus, γ_{k} is a kinematic hardening parameter, and $\varepsilon_{\mathrm{pe}}$ the effective plastic strain. Setting $\gamma_{\mathrm{k}} = 0$ 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 $\varepsilon_{\mathrm{back}}$ as proposed in (Ref. 8), which is related to the back stress as

$$\sigma_{\mathrm{back}} = \frac{2}{3}C_{\mathrm{k}}\varepsilon_{\mathrm{back}}$$

The nonlinear evolution for the back strain reads

$$\dot{\varepsilon}_{\mathrm{back}} = \dot{\varepsilon}_{\mathrm{p}} - \gamma_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{pe}}\varepsilon_{\mathrm{back}}$$

Chaboche Hardening Model

Chaboche (Ref. 8) proposed a nonlinear kinematic hardening model based on the superposition of N back stress tensors

$$\sigma_{\mathrm{back}} = \sum_{\mathrm{i}}^{N}\sigma_{\mathrm{back,i}}$$

each of these back stress tensors $\sigma_{\mathrm{back,i}}$ follows a nonlinear Frederick-Armstrong kinematic hardening rule

$$\dot{\sigma}_{\mathrm{back,i}} = \frac{2}{3}C_{\mathrm{i}}\dot{\varepsilon}_{\mathrm{p}} - \gamma_{\mathrm{i}}\dot{\varepsilon}_{\mathrm{pe}}\sigma_{\mathrm{back,i}}$$

Practitioners would normally select $\gamma_{\mathrm{k}} = 0$ for one of the back stress equations, thus recovering Prager's linear rule for that branch

$$\sigma_{\mathrm{back,0}} = \frac{2}{3}C_0\varepsilon_{\mathrm{p}}$$

The back stress tensor σ_{back} is then defined by the superposition of N back stress tensors

$$\sigma_{\mathrm{back}} = \frac{2}{3}C_0\varepsilon_{\mathrm{p}} + \sum_{\mathrm{i}=1}^{N}\sigma_{\mathrm{back,i}}$$

As done for Armstrong-Frederik kinematic hardening, the algorithm solves for the back strain tensors $\varepsilon_{\mathrm{back,i}}$ instead of the back stress tensors. The change of variables is

$$\sigma_{\mathrm{back,i}} = \frac{2}{3}C_{\mathrm{i}}\varepsilon_{\mathrm{back,i}}$$

and the nonlinear evolution for the back strain tensors reads

$$\dot{\varepsilon}_{\mathrm{back,i}} = \dot{\varepsilon}_{\mathrm{p}} - \gamma_{\mathrm{i}}\dot{\varepsilon}_{\mathrm{pe}}\varepsilon_{\mathrm{back,i}}$$

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 $\dot{\varepsilon}_{\mathrm{p}}$ 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

$$\dot{\varepsilon}_{\mathrm{p}} = \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma}$$

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 $\dot{\varepsilon}_p$) 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 $\dot{\varepsilon}_p$ 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*

$$\lambda \geq 0\,,\; F_y \leq 0 \;\text{ and }\; \lambda F_y \;=\; 0 \tag{3-25}$$

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 3-26 is solved together with the conditions in Equation 3-25.

$$\dot{\varepsilon}_p \;=\; \lambda\frac{\partial F_y}{\partial\sigma} \tag{3-26}$$

For a *non-associated* flow rule, the yield function does not coincide with the plastic potential, and together with the conditions in Equation 3-25, the rate in Equation 3-27 is solved for the plastic potential Q_p (often, a smoothed version of F_y).

$$\dot{\varepsilon}_p \;=\; \lambda\frac{\partial Q_p}{\partial\sigma} \tag{3-27}$$

The evolution of the plastic strain tensor $\dot{\varepsilon}_p$ (with either Equation 3-26 or Equation 3-27, plus the conditions in Equation 3-25) 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}:

$$-\frac{1}{2}\mathsf{L}(B_{\mathrm{el}}) = \lambda\frac{\partial\Phi}{\partial\tau}B_{\mathrm{el}} \qquad (3\text{-}28)$$

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

$$\lambda \ge 0\,,\ \Phi \le 0\ \text{ and }\ \lambda\Phi = 0$$

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 **d**, where **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

$$\mathsf{L}(B_{\mathrm{el}}) = F\dot{C}_{\mathrm{p}}^{-1}F^{T} \qquad (3\text{-}29)$$

By using Equation 3-28 and Equation 3-29, the either associated or non-associated plastic flow rule for large strains is written as (Ref. 10)

$$-\frac{1}{2}F\dot{C}_{\mathrm{p}}^{-1}F^{T} = \lambda\frac{\partial Q_{\mathrm{p}}}{\partial\tau}B_{\mathrm{el}} \qquad (3\text{-}30)$$

together with the Kuhn-Tucker conditions for the plastic multiplier λ and the yield function F_{y}

$$\lambda \ge 0\,,\ F_{\mathrm{y}} \le 0\ \text{ and }\ \lambda F_{\mathrm{y}} = 0 \qquad (3\text{-}31)$$

For the associated flow rule, the plastic potential and the yield surface coincide with each other ($Q_{\mathrm{p}} = F_{\mathrm{y}}$), and for the non-associated 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 $B_{\mathrm{el}} = FC_{\mathrm{p}}{}^{-1}F^{T}$. The flow rule then reads

$$\dot{C}_{\mathrm{p}}^{-1} = -2\lambda F^{-1}\frac{\partial Q_{\mathrm{p}}}{\partial \tau}FC_{\mathrm{p}}^{-1} \tag{3-32}$$

The plastic flow rule is then solved at Gauss points in the plastic element `elplastic` for the inverse of the plastic deformation gradient $F_{\mathrm{p}}{}^{-1}$, so that the variables in Equation 3-32 are replaced by

$$C_{\mathrm{p}}^{-1} = F_{\mathrm{p}}^{-1}F_{\mathrm{p}}^{-T},\ \dot{C}_{\mathrm{p}}^{-1} = \dot{F}_{\mathrm{p}}^{-1}F_{\mathrm{p}}^{-T} + F_{\mathrm{p}}^{-1}\dot{F}_{\mathrm{p}}^{-T} \text{ and } B_{\mathrm{el}} = FF_{\mathrm{p}}^{-1}F_{\mathrm{p}}^{-T}F^{T}$$

After integrating the flow rule in Equation 3-32, the plastic Green-Lagrange strain tensor is computed from the plastic deformation tensor

$$\varepsilon_{\mathrm{p}} = \frac{1}{2}(F_{\mathrm{p}}^{T}F_{\mathrm{p}} - I)$$

and the elastic Green-Lagrange strain tensor is computed from the elastic deformation gradient tensor $F_{\mathrm{el}} = FF_{\mathrm{p}}{}^{-1}$

$$\varepsilon_{\mathrm{el}} = \frac{1}{2}(F_{\mathrm{el}}^{T}F_{\mathrm{el}} - I)$$

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

$$\varepsilon_{\mathrm{p}} - \varepsilon_{\mathrm{p,\,old}} = \Lambda\frac{\partial Q_{\mathrm{p}}}{\partial \sigma}$$

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 3-32 is numerically solved with the so-called *exponential mapping* technique

$$C_{\mathrm{p}}^{-1} = F^{-1}\exp\left(-2\Lambda\frac{\partial Q_{\mathrm{p}}}{\partial\sigma}\right)FC_{\mathrm{p,\,old}}^{-1}$$

where $C_{\mathrm{p}}^{-1} = F_{\mathrm{p}}^{-1}F_{\mathrm{p}}^{-T}$ and $C_{\mathrm{p,\,old}}^{-1} = F_{\mathrm{p,\,old}}^{-1}F_{\mathrm{p,\,old}}^{-T}$.

For each Gauss point, the plastic state variables (ε_{p} or $F_{\mathrm{p}}{}^{-1}$, 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

$$\Lambda \ge 0, \qquad F_{\mathrm{y}} \le 0, \qquad \Lambda F_{\mathrm{y}} = 0$$

This is done as follows (Ref. 9):

1 Elastic-predictor: Try the elastic solution $\varepsilon_{\mathrm{p}} = \varepsilon_{\mathrm{p,old}}$ (or $F_{\mathrm{p}}^{-1} = F_{\mathrm{p,\,old}}^{-1}$) and $\Lambda = 0$. If this satisfies $F_{\mathrm{y}} \le 0$ it is done.

2 Plastic-corrector: If the elastic solution does not work (this is $F_{\mathrm{y}} > 0$), solve the nonlinear system consisting of the flow rule and the equation $F_{\mathrm{y}} = 0$ using a damped Newton method..

The numerical tolerance to fulfill the condition $F_{\mathrm{y}} = 0$ 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

$$\dot{W}_{\mathrm{p}} = \sigma{:}\dot{\varepsilon}_{\mathrm{p}} = \lambda\sigma{:}\frac{\partial Q_{\mathrm{p}}}{\partial\sigma}$$

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) 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 can be expressed in terms of the stress invariants as

$$F_y = \sqrt{J_2} + k_1 I_1^2 + k_2 I_1 + k_3 = 0 \tag{3-33}$$

here, k_1, k_2, and k_3 are material parameters.

This criterion can also be written (Ref. 17) in term of the uniaxial compressive strength f_c and the octahedral normal and shear stresses

$$\sigma_{oct} = I_1/3 \text{ and } \tau_{oct} = \sqrt{2J_2/3}$$

so Equation 3-33 simplifies to

$$F_y = \frac{\tau_{oct}}{f_c} - a + b\frac{\sigma_{oct}}{f_c} - c\frac{\sigma_{oct}^2}{f_c^2} = 0$$

Here, the parameters a, b, and c are obtained from the uniaxial compression, uniaxial tension, and biaxial compression tests, respectively. The octahedral normal stress σ_{oct} is considered positive when tensile, and f_{c} is taken positive.

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 is fitted to test data in the low compression range. The material is considered perfect elastoplastic (no hardening).

The original "three-parameter" Willam-Warnke failure criterion was defined as

$$F_{\mathrm{y}} = \sqrt{\frac{3}{5}}\frac{\tau_{\mathrm{oct}}}{f_{\mathrm{c}}} + r(\theta)\left(\left(\frac{1}{f_{\mathrm{t}}} - \frac{1}{f_{\mathrm{b}}}\right)\sigma_{\mathrm{oct}} - 1\right) = 0 \qquad (3\text{-}34)$$

where f_{c} is the uniaxial compressive strength, f_{t} is uniaxial tensile strength, and f_{b} is obtained from the biaxial compression test. All parameters are positive. The octahedral normal and shear stresses are defined as usual

$$\sigma_{\mathrm{oct}} = I_1/3\text{ , and }\tau_{\mathrm{oct}} = \sqrt{2J_2/3}$$

This can be written as

$$F_{\mathrm{y}} = \sqrt{J_2} + \sqrt{\frac{5}{2}}r(\theta)\left(\frac{1}{3\chi}I_1 - f_{\mathrm{c}}\right) = 0$$

with

$$\chi = f_{\mathrm{b}}f_{\mathrm{t}}/(f_{\mathrm{b}}f_{\mathrm{c}} - f_{\mathrm{t}}f_{\mathrm{c}})$$

The function $r(\theta)$ describes the segment of an ellipse on the octahedral plane when $0 \le \theta \le \pi/3$. By using the Lode angle θ, the dimensionless function $r(\theta)$ is defined as

$$r(\theta) = \frac{2r_c(r_c^2 - r_t^2)\cos\theta + r_c(2r_t - r_c)\sqrt{4(r_c^2 - r_t^2)\cos^2\theta + 5r_t^2 - 4r_c r_t}}{4(r_c^2 - r_t^2)\cos^2\theta + (r_c - 2r_t)^2}$$

Here, the tensile and compressive meridian r_{t} and r_{c} are defined in terms of the positive parameters f_{c}, f_{b}, and f_{t}:

$$r_t = \sqrt{\frac{6}{5}}\frac{f_{\mathrm{b}}f_{\mathrm{t}}}{2f_{\mathrm{b}}f_{\mathrm{c}}+f_{\mathrm{t}}f_{\mathrm{c}}}$$

$$r_c = \sqrt{\frac{6}{5}}\frac{f_{\mathrm{b}}f_{\mathrm{t}}}{3f_{\mathrm{b}}f_{\mathrm{t}}+f_{\mathrm{b}}f_{\mathrm{c}}-f_{\mathrm{t}}f_{\mathrm{c}}}$$

The function $r(\theta)$ can be interpreted as the friction angle which depends on the Lode angle (Ref. 10).

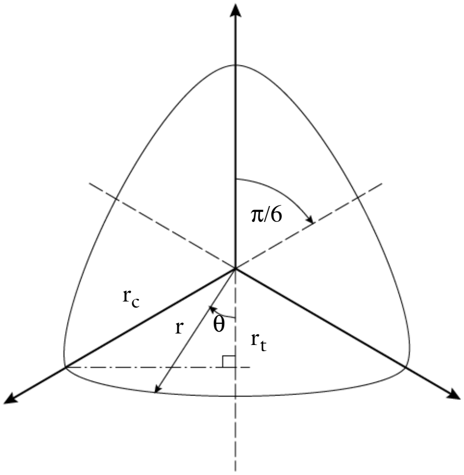


Figure 3-12: The deviatoric section of Willam-Warnke failure criterion.

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 is in agreement 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):

$$F_{\mathrm{y}} = \frac{a}{f_c}J_2 + \lambda(\theta)\sqrt{J_2} + bI_1 - f_c = 0$$

In this formulation, the parameters a>0, b>0, k_1>0, and k_2>0 are dimensionless, and f_c>0 is the uniaxial compressive strength of concrete (with a positive sign). The function $\lambda(\theta) > 0$ (dimensionless) is defined as

$$\lambda(\theta) = \begin{cases} k_1 \cos\left(\frac{1}{3}\operatorname{acos}(k_2\cos(3\theta))\right) & J_3 > 0 \\ k_1 \cos\left(\frac{\pi}{3} - \frac{1}{3}\operatorname{acos}(-k_2\cos(3\theta))\right) & J_3 < 0, \end{cases}$$

The parameter k_1 is called the *size factor*. The parameter k_2 (also called *shape factor*) is positive and bounded to $0 \le k_2 \le 1$(Ref. 17, Ref. 18).

Typical values for these parameters are obtained by curve-fitting the uniaxial compressive strength f_c, uniaxial tensile strength f_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 f_c, f_b, and f_t are positive.

TABLE 3-3: TYPICAL PARAMETER VALUES FOR OTTOSEN FAILURE CRITERION (Ref. 18).

f_t/f_c	a	b	k_1	k_2	l_t	l_c
0.08	1.8076	4.0962	14.4863	0.9914	14.4725	7.7834
0.10	1.2759	3.1962	11.7365	0.9801	11.7109	6.5315
0.12	0.9218	2.5969	9.9110	0.9647	9.8720	5.6979

The compressive and tensile meridians (as defined in the Willam-Warnke criterion) are

$$r_c = \frac{1}{\lambda_c} = \frac{1}{\lambda(\pi/3)}$$

$$r_t = \frac{1}{\lambda_t} = \frac{1}{\lambda(0)}$$

For concrete, the ratio

$$\lambda_c/\lambda_t = r_t/r_c$$

normally lies between 0.54~0.58.

The Ottosen criterion is equivalent to Drucker-Prager when $a = 0$ and λ = constant.

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, $0 \leq \mathrm{s} < 1$ 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

TABLE 3-4: CHARACTERISTIC VALUES FOR DIFFERENT ROCK TYPES

m	ROCK TYPE
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 $0 \le \theta \le \pi/3$, so

$$F_\mathrm{y} = 2\sqrt{J_2}\sin\left(\theta + \frac{\pi}{3}\right) - \sigma_c\sqrt{s - m\frac{\sigma_1}{\sigma_c}} = 0$$

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 $0 \le \theta \le \pi/3$ this equals

$$F_\mathrm{y} = 2\sqrt{J_2}\sin\left(\theta + \frac{\pi}{3}\right) - \sigma_c\left(s - m_b\frac{\sigma_1}{\sigma_c}\right)^a = 0$$

where $\sigma_1 \ge \sigma_2 \ge \sigma_3$ 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:

$$m_b = m_i\exp\left(\frac{\mathrm{GSI} - 100}{28 - 14D}\right)$$

s and a are positive parameters for the rock mass given by the following relationships:

$$s = \exp\left(\frac{\mathrm{GSI} - 100}{9 - 3D}\right)$$

$$a = \frac{1}{2} + \frac{1}{6}\left(\exp\left(\frac{-\mathrm{GSI}}{15}\right) - \exp\left(\frac{-20}{3}\right)\right)$$

The *disturbance factor* D was introduced to account for the effects of stress relaxation and blast damage, and it varies from 0 for undisturbed in-situ rock masses to 1 for very damaged rock masses.

TABLE 3-5: 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

Cam-Clay Material Model

ABOUT THE CAM-CLAY MATERIAL MODEL

The Cam-clay material model was developed at the University of Cambridge in the 1970s, and since then it has experienced different modifications. The *modified Cam-clay model* is the most commonly used due to the smooth yield surface, and it is the one implemented in the Geomechanics Module.

The 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 these models (see Ref. 13, Ref. 14, and Ref. 15).

The yield function is written in terms of the variables

$$q = \sqrt{3J_2(\sigma)} \text{ and } p = -I_1(\sigma)/3$$

Following the Structural Mechanics Module sign convention:

$$F_\mathrm{y} = q^2 + M^2(p - p_c)p = 0$$

This is an ellipse in p-q plane, with a cross section independent of Lode angle and smooth for differentiation. Note that p, q, and p_c are always positive variables.

The material parameter $M > 0$ defines the slope of a line in the p-q space called *critical state line*, and it can be related to the angle of internal friction ϕ in the Mohr-Coulomb criterion

$$M = \frac{6\sin(\phi)}{3 - \sin(\phi)}$$

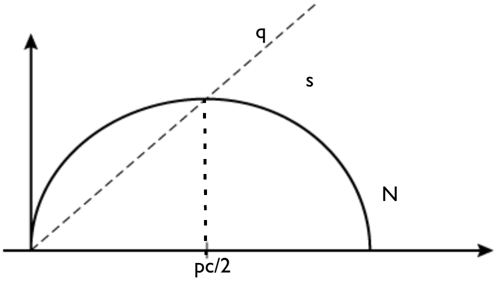


Figure 3-13: Modified Cam-clay surface in the pq-plane. The ellipse circumscribes a nonlinear elastic region.

In the Cam-clay model, hardening is controlled by the *consolidation pressure* p_c, which depends exponentially on the volumetric plastic strain $\varepsilon_{\mathrm{pl,vol}}$.

$$p_c = p_{c0}e^{-B_{pl}\varepsilon_{pl,vol}} \tag{3-35}$$

The volumetric plastic strain is available in the variable `solid.epvol` and the consolidation pressure in the variable `solid.Pc`.

Here, the parameter p_{c0} is the initial consolidation pressure, and the exponent B_{pl} is a parameter which depends on the initial void ratio e_0, the *swelling index* κ, and the *compression index* λ:

$$B_{\mathrm{pl}} = \frac{1+e_0}{\lambda-\kappa}$$

The initial void ratio, the compression index, and the swelling index are all positive parameters and must fulfill

$$0<\kappa<\lambda\text{, so }B_{\mathrm{pl}}>0$$

The void ratio e is the ratio between pore volume and solid volume. It can be written in terms of the porosity ϕ as $e = \phi/(1-\phi)$.

In this formulation, the compression index λ is the slope of the *virgin isotropic consolidation line*, and κ is the slope of the *rebound-reloading line* (also called loading-reloading line) in the e versus $\ln(p)$ plane.

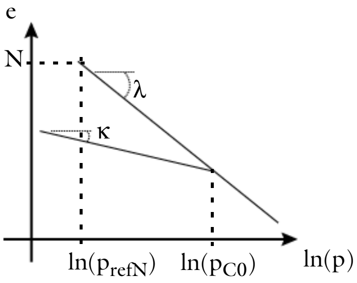


Figure 3-14: Slopes of the virgin isotropic consolidation line and rebound-reloading line in the e vs. ln(p) plane. The reference void ratio N is measured at the reference pressure p_{refN}.

If an Initial Stress and Strain feature subnode 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 BEHAVIOR

The stress-strain relation beyond the elastic range is of great importance in soil mechanics. For additive decomposition of strains, Cauchy's stress tensor is written as

$$\sigma - \sigma_0 = \mathbf{C}:(\varepsilon - \varepsilon_{\mathrm{inel}})$$

Here, σ is the Cauchy stress tensor, ε is the total strain tensor, $\varepsilon_{\mathrm{inel}}$ is the inelastic strain tensor, σ_0 is the initial stress tensor, and $\mathbf{C}$ is the fourth-order elasticity tensor.

For a linear elastic material, the trace of the Cauchy's stress tensor is linearly related to the volumetric elastic strain (the trace of the elastic strain tensor) by the elastic bulk modulus

$$p = -I_1(\sigma)/3 = p_0 - K\varepsilon_{\mathrm{el,vol}}$$

here $p_0 = -\mathrm{trace}(\sigma_0)/3$ is the trace of the initial stress tensor σ_0, and K is the bulk modulus, a constant parameter independent of the stress or strain.

The elastic volumetric strain is available in the variable `solid.eelvol`.

The modified Cam-clay model introduces a nonlinear relation between stress and volumetric elastic strain

$$p - p_0 = -K_0 e^{-B_{el}\varepsilon_{el,vol}} \text{ with } B_{\mathrm{el}} = \frac{1+e_0}{\kappa}$$

and K_0 a reference bulk modulus. This formulation gives a tangent bulk modulus $K_T = -B_{\mathrm{el}}(p - p_0)$. The reference bulk modulus is calculated from the initial consolidation pressure p_{c0}, and the void ratio at reference pressure N.

HARDENING AND SOFTENING

The yield surface for the modified Cam-clay model reads

$$F_{\mathrm{y}} = q^2 + M^2(p - p_c)p = 0$$

The associated flow rule ($Q_{\mathrm{p}} = F_{\mathrm{y}}$) and the yield surface written in terms of these two invariants, $F_{\mathrm{y}}(I_1, J_2)$, gives a rate equation for the plastic strain tensor calculated from the derivatives of F_{y} with respect to the stress tensor σ

$$\dot{\varepsilon}_{\mathrm{p}} = \lambda_{\mathrm{p}}\frac{\partial Q_{\mathrm{p}}}{\partial\sigma} = \lambda_{\mathrm{p}}\left(\frac{\partial Q_{\mathrm{p}}}{\partial I_1}\frac{\partial I_1}{\partial\sigma} + \frac{\partial Q_{\mathrm{p}}}{\partial J_2}\frac{\partial J_2}{\partial\sigma}\right)$$

Here, λ_{p} means the plastic multiplier, see Plastic Flow for Small Strains and Isotropic Hardening.

The plastic strain rate tensor $\dot{\varepsilon}_{\mathrm{p}}$ includes both deviatoric and isotropic parts. Note that

$$\partial I_1/\partial\sigma = \mathbf{I} \quad \text{and} \quad \partial J_2/\partial\sigma = \mathrm{dev}(\sigma)$$

These relations can be used for writing the plastic flow as

$$\dot{\varepsilon}_\mathrm{p} = \lambda_\mathrm{p}\left(-\frac{1}{3}\frac{\partial Q_\mathrm{p}}{\partial p}\mathbf{I} + \frac{\partial Q_\mathrm{p}}{\partial q}\frac{3}{2q}\mathrm{dev}(\sigma)\right) = \lambda_\mathrm{p}\left(-\frac{1}{3}M^2(2p-p_c)\mathbf{I} + 3\mathrm{dev}(\sigma)\right)$$

The trace of the plastic strain rate tensor (the *volumetric plastic strain rate* $\dot{\varepsilon}_\mathrm{pvol}$) then reads

$$\dot{\varepsilon}_\mathrm{pvol} = \mathrm{trace}(\dot{\varepsilon}_\mathrm{p}) = \lambda_\mathrm{p}(-M^2(2p-p_c))$$

This relation explains the reason why there is *isotropic hardening* for $p > p_c/2$ and *isotropic softening* for $p < p_c/2$. So the volumetric plastic strain can either increase or decrease.

In the Cam-clay model, the hardening is controlled by the consolidation pressure variable $p_c(\varepsilon_\mathrm{pvol})$ as a function of volumetric plastic strain, as written in Equation 3-35. Hardening introduces changes in the shape of the Cam-clay ellipse, since its major semi-axis depends on the value of p_c.

INCLUDING PORE FLUID PRESSURE

When a pore fluid pressure p_f is added to the Cam-clay material, the yield surface is shifted on the p axis

$$F_\mathrm{y} = q^2 + M^2(p - p_\mathrm{f} - p_c)(p - p_\mathrm{f}) = 0$$

The quantity $p - p_\mathrm{f}$ is normally regarded as the *effective pressure*, or effective stress, which should not be confused with von Mises stress.

See also the description of the Cam-Clay Material material model in the Solid Mechanics interface documentation.

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:

$$\dot{\varepsilon}_{\mathrm{cr}} = F_{\mathrm{cr}}(\sigma, T, t)$$

It is often possible to separate these effects as shown in this equation:

$$F_{\mathrm{cr}}(\sigma, t, T) = f_1(\sigma) f_2(T) f_3(t)$$

Experimental data shows three types of behavior for the creep strain rate at constant stress as function of time. Researchers normally subdivide the creep curve into three regimes, based on the fact that many different materials show similar responses:

- In the initial *primary creep* regime (also called *transient creep*) the creep strain rate decreases with time to a minimum steady-state value.
- In the *secondary creep* regime the creep strain rate is almost constant. This is also called *steady-state* creep.
- In the *tertiary creep* regime the creep strain increases with time until a failure occurs.

When this distinction is assumed, the total creep rate can be additively split into primary, secondary, and tertiary creep rates

$$\dot{\varepsilon}_{\mathrm{cr}} = F_{\mathrm{cr1}} + F_{\mathrm{cr2}} + F_{\mathrm{cr3}}$$

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.

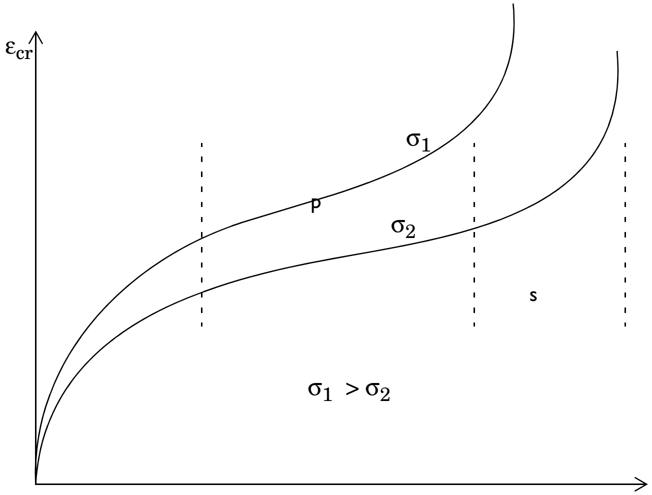


Figure 3-15: Uniaxial creep as a function of logarithmic time.

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)
- Navarro-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 the 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:

$$\dot{\varepsilon}_{\mathrm{cr}} = \eta\frac{\partial Q_{\mathrm{cr}}}{\partial\sigma} \text{ and } \eta > 0$$

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

$$\frac{\partial Q_{\mathrm{cr}}}{\partial\sigma} = \frac{\partial Q_{\mathrm{cr}}}{\partial I_1}\mathbf{I}$$

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

$$\frac{\partial Q_{\mathrm{cr}}}{\partial\sigma} = \frac{\partial Q_{\mathrm{cr}}}{\partial J_2}\mathrm{dev}(\sigma)$$

This is equivalent to that the creep potential would depend on the effective stress $\sigma_{\mathrm{e}} = \sqrt{3J_2}$.

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

$$\dot{\varepsilon}_{cr} = \frac{1}{3}F_{cr}\mathbf{I}$$

so that the creep rate tensor is a diagonal tensor. The trace of the creep rate tensor, the *volumetric creep strain rate*, equals the user input F_{cr}

$$\mathrm{trace}(\dot{\varepsilon}_{cr}) = F_{cr} \tag{3-36}$$

The creep rate, F_{cr}, 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

$$\dot{\varepsilon}_{cr} = F_{cr}\mathbf{n}^D$$

Here, $\mathbf{n}^D$ is a deviatoric tensor coaxial to the stress tensor.

The creep rate, F_{cr}, normally depends on the second deviatoric invariant of the stress $J_2(\sigma)$ 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

$$\mathbf{n}^D = \frac{3}{2}\frac{\mathrm{dev}(\sigma)}{\sigma_e} \tag{3-37}$$

The resulting creep strain rate tensor is also deviatoric, since trace $(\mathbf{n}^D) = 0$

$$\mathrm{trace}(\dot{\varepsilon}_{cr}) = F_{cr}\mathrm{trace}(\mathbf{n}^D) = 0$$

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 F_{cr}

$$\dot{\varepsilon}_{cr,e} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{cr}:\dot{\varepsilon}_{cr}} = |F_{cr}|$$

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 rate is calculated by solving the rate equation

$$\dot{\varepsilon}_{cr} = F_{cr}$$

where F_{cr} is a user-defined symmetric tensor field.

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:

$$\dot{\varepsilon}_{cr} \propto \sigma_e{}^n$$

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

$$\dot{\varepsilon}_{cr} \propto e^{-Q/RT}$$

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

$$\dot{\varepsilon}_{cr} = A\left(\frac{\sigma_e}{\sigma_{ref}}\right)^n e^{-\frac{Q}{RT}}\mathbf{n}^D \tag{3-38}$$

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 3-37.

- 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

$$\varepsilon_{\mathrm{cr}} \propto {\sigma_{\mathrm{e}}}^n t^m$$

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.

$$\dot{\varepsilon}_{\mathrm{cr}} \propto {\sigma_{\mathrm{e}}}^n m t^{m-1}$$

Norton-Bailey creep is a deviatoric temperature-dependent creep model, furbished 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

$$\dot{\varepsilon}_{\mathrm{cr}} = F_{\mathrm{cr}} m \left(\frac{t + t_{\mathrm{shift}}}{t_{\mathrm{ref}}}\right)^{m-1} \mathbf{n}^D \qquad (3\text{-}39)$$

where $\mathbf{n}^D$ is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-37, and F_{cr} is expressed as in the Norton model:

$$F_{\mathrm{cr}} = A\left(\frac{\sigma_{\mathrm{e}}}{\sigma_{\mathrm{ref}}}\right)^n e^{-\frac{Q}{RT}} \qquad (3\text{-}40)$$

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

$$\dot{\varepsilon}_{\mathrm{cr}} = F_{\mathrm{cr}} m \left(\frac{\varepsilon_{\mathrm{cr,e}} + \varepsilon_{\mathrm{shift}}}{t_{\mathrm{ref}} F_{\mathrm{cr}}}\right)^{\frac{m-1}{m}} \mathbf{n}^D \qquad (3\text{-}41)$$

where $\varepsilon_{\mathrm{cr,e}}$ is the effective creep strain, and $\varepsilon_{\mathrm{shift}}$ is the effective creep strain shift.

The time and frequency shifts in Equation 3-39 and Equation 3-41 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

$$\dot{\varepsilon}_{\mathrm{cr}} \propto e^{\alpha\sigma_{\mathrm{e}}}$$

Garofalo showed (Ref. 8, Ref. 9) that the power-law and exponential creep are limiting cases for the general empirical expression

$$\dot{\varepsilon}_{\mathrm{cr}} \propto (\sinh(\alpha\sigma_{\mathrm{e}}))^n$$

This equation reduces to a power-law (Norton law) for $\alpha\sigma_{\mathrm{e}} < 0.8$ and approaches exponential creep for $\alpha\sigma_{\mathrm{e}} > 1.2$, 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

$$\dot{\varepsilon}_{\mathrm{cr}} \propto e^{-Q/RT}$$

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

$$\dot{\varepsilon}_{\mathrm{cr}} = A\left(\sinh\left(\frac{\sigma_{\mathrm{e}}}{\sigma_{\mathrm{ref}}}\right)\right)^{n} e^{-\frac{Q}{RT}}\mathbf{n}^{D}$$

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 3-37.

- 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.

NAVARRO-HERRING CREEP (DIFFUSIONAL CREEP)

At low stress levels and high temperatures, Navarro and Herring (Ref. 6, Ref. 7) independently derived an expression for the creep rate as a function of atomic diffusion

$$\dot{\varepsilon}_{\mathrm{cr}} = \frac{7D_{\mathrm{v}}b^3}{k_{\mathrm{B}}Td^2}\sigma_{\mathrm{e}}\mathbf{n}^{D}$$

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 3-37.

See also the description of the Navarro-Herring material model in the Solid Mechanics interface documentation.

COBLE CREEP (DIFFUSIONAL CREEP)

Coble creep (Ref. 6, Ref. 7) is closely related to Navarro-Herring creep but takes into account the ionic diffusivity along grain boundaries D_{gb}

$$\dot{\varepsilon}_{\mathrm{cr}} = \frac{50D_{\mathrm{gb}}b^4}{k_{\mathrm{B}}Td^3}\sigma_{\mathrm{e}}\mathbf{n}^D$$

Coble creep is more sensitive to grain size than Navarro-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)

$$\dot{\varepsilon}_{\mathrm{cr}} = \frac{Db}{k_{\mathrm{B}}T}\sigma_{\mathrm{ref}}\left(\frac{\sigma_{\mathrm{e}}}{\sigma_{\mathrm{ref}}}\right)^n\mathbf{n}^D$$

where and $\mathbf{n}^D$ is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-37. 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)

$$\dot{\varepsilon}_{\mathrm{cr}} \propto \frac{DGb}{k_BT}\left(\frac{\sigma_{\mathrm{e}}}{G}\right)^n\left(\frac{b}{d}\right)^p e^{-Q/RT}$$

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

$$\dot{\varepsilon}_{\mathrm{cr}} = F_{\mathrm{cr}}\mathbf{n}^D$$

where $\mathbf{n}^D$ is a deviatoric tensor coaxial to the stress tensor as defined in Equation 3-37, and the creep rate is calculated from

$$F_{\mathrm{cr}} = Ae^{-Q/RT}\left[\sinh\left(\xi\frac{\sigma_{\mathrm{e}}}{s_{\mathrm{a}}}\right)\right]^{\frac{1}{m}}$$

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

$$\dot{s}_{\mathrm{a}} = h_0\left|1-\frac{s_{\mathrm{a}}}{s_{\mathrm{a}}^*}\right|^{a-1}\left(1-\frac{s_{\mathrm{a}}}{s_{\mathrm{a}}^*}\right)F_{\mathrm{cr}}$$

with the initial condition $s_{\mathrm{a}}(0) = s_{\mathrm{init}}$. 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

$$s_{\mathrm{a}}^* = s_0\left(\frac{F_{\mathrm{cr}}}{A}e^{-Q/RT}\right)^n$$

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

$$\dot{\varepsilon}_{\mathrm{vp}} = A\left\langle\frac{F_{\mathrm{y}}}{\sigma_{\mathrm{ref}}}\right\rangle^{n}\mathbf{n}^{D}$$

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

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

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}

$$\mathbf{n}^{D} = \frac{\partial Q_{\mathrm{p}}}{\partial\sigma}$$

When von Misses effective stress is used, the associated flow rule reads $Q_{\mathrm{p}} = F_{\mathrm{y}}$, and the deviatoric tensor $\mathbf{n}^D$ is defined as done for deviatoric creep

$$\mathbf{n}^{D} = \frac{3}{2}\frac{\mathrm{dev}(\sigma)}{\sigma_{\mathrm{mises}}} \tag{3-42}$$

Given the property

$$\mathbf{n}^{D}:\mathbf{n}^{D} = \frac{3}{2}$$

the effective viscoplastic strain rate is equivalent to

$$\dot{\varepsilon}_{\mathrm{vpe}} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{\mathrm{vp}}:\dot{\varepsilon}_{\mathrm{vp}}} = A\left\langle\frac{F_{\mathrm{y}}}{\sigma_{\mathrm{ref}}}\right\rangle^{n}$$

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

$$\dot{\varepsilon}_{\mathrm{vp}} = \frac{A}{\sigma_{\mathrm{ref}}}\langle F_{\mathrm{y}}\rangle \mathbf{n}^{D}$$

Some authors denote the viscosity as the quotient $\eta = \sigma_{\mathrm{ref}}/A$.

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

$$\dot{W}_{\mathrm{cdr}} = \sigma{:}\dot{\varepsilon}_{\mathrm{cr}}$$

In case many creep sub-nodes are added, the creep dissipation rate density is calculated from the cumulative creep strain rate tensor $\dot{\varepsilon}_{\mathrm{cr}}$.

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 that the model units. Since many creep models contain the stress or time raised to a non-integer power, such conversions are error prone.

Norton law

Material data for a Norton law is often available as the parameters A_N and n in the equation

$$\dot{\varepsilon}_\mathrm{cr} = A_\mathrm{N}\sigma_\mathrm{e}^{\ \ n}$$

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 3-38) 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 $(\mathrm{time})^{-1}$, whereas the physical dimension of A_N is $(\mathrm{stress})^{-\mathrm{n}}(\mathrm{time})^{-1}$.

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}/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 `1e-7[1/h]`.

Example

Assume that a carbon steel has the following two equivalent descriptions of its creep properties at a certain temperature:

- σ_c7 = 70 MPa, and stress exponent n = 4.5.
- A_N = $4.98\cdot10^{-16}$ with respect to units MPa and hours, and stress exponent n = 4.5.

In the first case, enter:

- σ_ref as `70[MPa]`

- n as 4.5
- A as `1e-7[1/h]`.

In the second case, enter:

- σ_{ref} as `1[MPa]`
- n as 4.5
- A as `4.98e-16[1/h]`

These two sets of data describe the same material.

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

$$\varepsilon_{\mathrm{cr}} = A_{\mathrm{NB}}\sigma_{\mathrm{e}}{}^{n}t^{m}$$

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 3-39 and Equation 3-40) 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

$$\dot{\varepsilon}_{\mathrm{cr}} \propto (\sinh(\alpha\sigma_{\mathrm{e}}))^{n}$$

Comparing with the expression in COMSOL Multiphysics,

$$\dot{\varepsilon}_{\mathrm{cr}} \propto \left(\sinh\left(\frac{\sigma_{\mathrm{e}}}{\sigma_{\mathrm{ref}}}\right)\right)^{n}$$

it is evident that the reference stress should be chosen as

$$\sigma_{\mathrm{ref}} = \frac{1}{\alpha}$$

In this case, there is no arbitrariness in the choice of σ_{ref}, since α is an actual material parameter.

Piezoelectric Material

THE PIEZOELECTRIC EFFECT

The piezoelectric effect manifests itself as a transfer of electric to mechanical energy and vice versa. It is present in many crystalline materials, while some materials such as quartz, Rochelle salt, and lead titanate zirconate ceramics display the phenomenon strongly enough for it to be of practical use.

The *direct piezoelectric effect* consists of an electric polarization in a fixed direction when the piezoelectric crystal is deformed. The polarization is proportional to the deformation and causes an electric potential difference over the crystal.

The *inverse piezoelectric effect*, on the other hand, constitutes the opposite of the direct effect. This means that an applied potential difference induces a deformation of the crystal.

PIEZOELECTRIC CONSTITUTIVE RELATIONS

It is possible to express the relation between the stress, strain, electric field, and electric displacement field in either a *stress-charge form* or *strain-charge form*:

Stress-Charge

$$\begin{aligned} T &= c_E S - e^T \mathbf{E} \\ \mathbf{D} &= eS + \varepsilon_S \mathbf{E} \end{aligned}$$

Strain-Charge

$$\begin{aligned} S &= s_E T + d^T \mathbf{E} \\ \mathbf{D} &= dT + \varepsilon_T \mathbf{E} \end{aligned}$$

In the above relations, the naming convention used in piezoelectricity theory is assumed, so that the structural strain is denoted by S, and the stress is denoted by T. Thus, the naming convention differs in piezoelectricity theory compared to structural mechanics theory.

> **!**
>
> The Piezoelectric Material uses the structural mechanics nomenclature. The strain is named ε (instead of $\mathbf{S}$) 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

$$\begin{aligned} \sigma &= c_E\varepsilon - e^T\mathbf{E} \\ \mathbf{D} &= e\varepsilon + \varepsilon_0\varepsilon_{\mathrm{r}S}\mathbf{E} \end{aligned} \tag{3-43}$$

The Stress-Charge form is always used in the variational formulation (weak equation form) which COMSOL Multiphysics uses for discretization and computation.

STRAIN-CHARGE

$$\begin{aligned} \varepsilon &= s_E\sigma + d^T\mathbf{E} \\ \mathbf{D} &= d\sigma + \varepsilon_0\varepsilon_{\mathrm{r}T}\mathbf{E} \end{aligned}$$

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:

$$\begin{aligned} c_E &= s_E^{-1} \\ e &= d\,s_E^{-1} \\ \varepsilon_S = \varepsilon_0\varepsilon_{\mathrm{r}S} &= \varepsilon_0\varepsilon_{\mathrm{r}T} - d\,s_E^{-1}\,d^T \end{aligned}$$

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 3-51 with the charge conservation equation of Electrostatics,

$$\nabla\cdot\mathbf{D} = \rho_V \tag{3-44}$$

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 3-44 and Equation 3-51, the constitutive relations Equation 3-43 are used, which makes the resulting system of equations closed. The dependent variables are the structural displacement vector **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:

$$\mathbf{u} = \hat{\mathbf{u}}\exp[i\mathbf{k}(\mathbf{r}+\mathbf{c}t)]$$
$$V = \hat{V}\exp[i\mathbf{k}(\mathbf{r}+\mathbf{c}t)]$$

where **k** is the wave number vector that determines the direction of the wave propagation, and **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

$$u = \hat{u}\exp[ik_X(X+c_Xt)]$$
$$v = 0$$
$$w = 0$$
$$V = \hat{V}\exp[ik_X(X+c_Xt)]$$

The corresponding pressure wave speed is given by

$$c_{pX} = \frac{1}{\rho}\left[c_{E,11} + \frac{e_{X1}^2}{\varepsilon_0\varepsilon_{rS,XX}}\right]^{1/2}$$

The shear wave propagation in the X axis direction and with XY plane polarization is a solution such that

$$u = 0$$

$$v = \hat{v}\exp[ik_X(X + c_X t)]$$

$$w = 0$$

$$V = \hat{V}\exp[ik_X(X + c_X t)]$$

and the corresponding wave speed is computed as

$$c_{sXY} = \frac{1}{\rho}\left[c_{E,66} + \frac{e_{X6}^2}{\varepsilon_0 \varepsilon_{rS,XX}}\right]^{1/2}$$

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

$$\tilde{X} = X(1 \pm j\eta_X)$$

See also the same references for an explanation of the sign convention.

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.

Piezoelectric Losses

INITIAL STRESS, STRAIN, AND ELECTRIC DISPLACEMENT

Using the functionality available under the Piezoelectric Material feature and Solid Mechanics interface, one can define initial stress (S_0), initial strain (ε_0), and remanent electric displacement ($\mathbf{D_r}$) for models. In the constitutive relation for piezoelectric material these additions appear in the stress-charge formulation:

$$\begin{aligned}\sigma &= c_E(\varepsilon - \varepsilon_0) - e^T\mathbf{E} + \sigma_0 \\ \mathbf{D} &= e(\varepsilon - \varepsilon_0) + \varepsilon_{0,\,\mathrm{vac}}\varepsilon_{\mathrm{r}S}\mathbf{E} + \mathbf{D}_\mathrm{r}\end{aligned}$$

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.

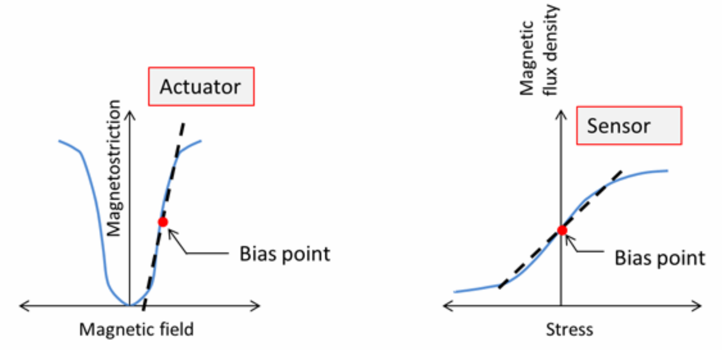
Magnetostrictive Material

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 S, strain ε, magnetic field **H**, and magnetic flux density **B** in either a *stress-magnetization form* or *strain-magnetization form*:

Stress-Magnetization

$$\begin{aligned} \mathrm{S} &= c_{\mathrm{H}}\varepsilon - e_{\mathrm{HS}}^{T}\mathbf{H} \\ \mathbf{B} &= e_{\mathrm{HS}}\varepsilon_{\mathrm{el}} + \mu_0\mu_{\mathrm{rS}}\mathbf{H} \end{aligned}$$

Strain-Magnetization

$$\begin{aligned} \varepsilon &= s_{\mathrm{H}}\mathrm{S} + d_{\mathrm{HT}}^{T}\mathbf{H} \\ \mathbf{B} &= d_{\mathrm{HT}}\mathrm{S} + \mu_0\mu_{\mathrm{rT}}\mathbf{H} \end{aligned}$$

where μ_0 is the magnetic permeability of free space, c_{H} and s_{H} are respectively the stiffness and compliance matrices measured at constant magnetic field, and μ_{rS} and μ_{rT} are the relative magnetic permeabilities measured at constant strain and constant stress, respectively. The matrices d_{HT} and e_{HS} are called piezomagnetic coupling matrices.

In COMSOL Multiphysics, both constitutive forms can be used; simply select one, and the software makes all necessary transformations. The following equations transform strain-magnetization material data to stress-magnetization data:

$$
\begin{aligned}
c_{\mathrm{H}} &= s_{\mathrm{H}}^{-1} \\
e_{\mathrm{HS}} &= d_{\mathrm{HT}} s_{\mathrm{H}}^{-1} \\
\mu_{\mathrm{rS}} &= \mu_{\mathrm{rT}} - \frac{1}{\mu_0} d_{\mathrm{HT}} s_E^{-1} d_{\mathrm{HT}}^T
\end{aligned}
$$

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:

$$
\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & 0 & 0 & 0 \\ s_{12} & s_{11} & s_{13} & 0 & 0 & 0 \\ s_{13} & s_{13} & s_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(s_{11}-s_{12}) \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{23} \\ S_{13} \\ S_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & -\frac{1}{2}d_{33} \\ 0 & 0 & d_{33} \\ 0 & d_{15} & 0 \\ d_{15} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}
$$

$$
\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{15} & 0 & 0 \\ -\frac{1}{2}d_{33} & -\frac{1}{2}d_{33} & d_{33} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{23} \\ S_{13} \\ S_{12} \end{bmatrix} + \mu_0 \begin{bmatrix} \mu_{11} & 0 & 0 \\ 0 & \mu_{11} & 0 \\ 0 & 0 & \mu_{33} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix}
$$

The following material data corresponds to Terfenol-D at 100 kA/m bias and 30 MPa prestress (Ref. 5):

$$s_{\mathrm{H}} = 10^{-11}\cdot\begin{bmatrix} 4.4 & -1.1 & -1.65 & 0 & 0 & 0 \\ -1.1 & 4.4 & -1.65 & 0 & 0 & 0 \\ -1.65 & -1.65 & 3.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 \end{bmatrix}[1/\mathrm{Pa}]$$

$$d_{\mathrm{HT}} = 10^{-9}\cdot\begin{bmatrix} 0 & 0 & 0 & 0 & 16.5 & 0 \\ 0 & 0 & 0 & 16.5 & 0 & 0 \\ -4.3 & -4.3 & 8.6 & 0 & 0 & 0 \end{bmatrix}[\mathrm{m/A}] \qquad \mu_{\mathrm{rT}} = \begin{bmatrix} 8.1 & 0 & 0 \\ 0 & 8.1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

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:

$$\varepsilon_{\mathrm{me}} = \frac{3}{2}\left[\lambda_{100}\left(\mathbf{m}\otimes\mathbf{m} - \frac{1}{3}I\right) + (\lambda_{111} - \lambda_{100})\sum_{i\neq j} m_i m_j(\mathbf{e}_i \otimes \mathbf{e}_j)\right] \qquad (3\text{-}45)$$

where $(\mathbf{m}\otimes\mathbf{m})_{ij} = m_i m_j$. Note that the magnetostrictive strain is represented by a deviatoric tensor, i.e. $\mathrm{tr}(\varepsilon_{\mathrm{me}}) = 0$. 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 3-45, one gets

$$\lambda = \frac{3}{2}\lambda_{100}\left(m_1^2\beta_1^2 + m_2^2\beta_2^2 + m_3^2\beta_3^2 - \frac{1}{3}\right) + \\ 3\lambda_{111}(m_1 m_2 \beta_1 \beta_2 + m_2 m_3 \beta_2 \beta_3 + m_1 m_3 \beta_1 \beta_3) \tag{3-46}$$

When both magnetization and measurement direction are parallel to the same crystal direction [100], one has $m_1 = \beta_1 = 1$ and all other components are zero, so that

$$\lambda = \lambda_{||} = \lambda_{100}$$

In a similar way for the [111] direction, one has for all components $m_i = \beta_i = 1/\sqrt{3}$ and $\lambda = \lambda_{111}$.

If the strain is measured in [100] direction, while all the magnetization vectors are aligned perpendicular to it, one has only the following two nonzero components: $m_2 = \beta_1 = 1$ and consequently:

$$\lambda = \lambda_{\perp} = -\frac{1}{2}\lambda_{100}$$

In many applications, such alignment of the domains is achieved by applying a compressive prestress. Thus, the maximum usable magnetostriction is achieved via a 90 degree rotation of the domains

$$\lambda_{||} - \lambda_{\perp} = \frac{3}{2}\lambda_{100}$$

For an isotropic material, $\lambda_{100} = \lambda_{111} = \lambda_s$, and Equation 3-45 becomes

$$\varepsilon_{\mathrm{me}} = \frac{3}{2}\lambda_s\left(\mathbf{m}\otimes\mathbf{m} - \frac{1}{3}I\right)$$

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:

$$\varepsilon_{\mathrm{me}} = \frac{3}{2}\frac{1}{M_s^2}\left[\lambda_{100}\mathrm{dev}(\mathbf{M}\otimes\mathbf{M}) + (\lambda_{111} - \lambda_{100})\sum_{i\neq j} M_i M_j(\mathbf{e}_i \otimes \mathbf{e}_j)\right] \tag{3-47}$$

The strain field is deviatoric, and Equation 3-47 exhibits the same properties as Equation 3-45 at saturation, i.e. when $|\mathbf{M}| \to M_s$. Equation 3-46 is replaced by

$$\lambda = \frac{3}{2}\frac{\lambda_{100}}{M_s^2}\left[M_1^2\left(\beta_1^2 - \frac{1}{3}\right) + M_2^2\left(\beta_2^2 - \frac{1}{3}\right) + M_3^2\left(\beta_3^2 - \frac{1}{3}\right)\right] +$$
$$3\frac{\lambda_{111}}{M_s^2}(M_1M_2\beta_1\beta_2 + M_2M_3\beta_2\beta_3 + M_1M_3\beta_1\beta_3)$$

Note that the strain vanishes when $|\mathbf{M}| \to 0$, which makes the model applicable in the whole range from full demagnetization to saturation.

For isotropic materials, the magnetostrictive strain is modeled as the following quadratic isotropic form of the magnetization field (Ref. 3):

$$\varepsilon_{\mathrm{me}} = \frac{3}{2}\frac{\lambda_s}{M_s^2}\mathrm{dev}(\mathbf{M}\otimes\mathbf{M}) \tag{3-48}$$

The stress in the magnetostrictive material is modeled as

$$\mathrm{S} = \mathrm{c_H}[\varepsilon - \varepsilon_{\mathrm{me}}(\mathbf{M})]$$

For isotropic materials, the stiffness matrix $\mathrm{c_H}$ 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 3-48, one can derived a linear response around a given bias state characterized by a pre-magnetization 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

$$\varepsilon_{\mathrm{me}} = \frac{\lambda_s}{M_s^2}\begin{bmatrix} 2M_{0,1} & -M_{0,2} & -M_{0,3} \\ -M_{0,1} & 2M_{0,2} & -M_{0,3} \\ -M_{0,1} & -M_{0,2} & 2M_{0,3} \\ 0 & 3M_{0,3} & 3M_{0,2} \\ 3M_{0,3} & 0 & 3M_{0,1} \\ 3M_{0,2} & 3M_{0,1} & 0 \end{bmatrix}\mathbf{M}_1$$

If one assumes a unidirectional state, for example

$$\mathbf{M}_0 = [0, 0, M_0]$$

it will further simplify into

$$\varepsilon_{\mathrm{me}} = \frac{\lambda_s M_0}{M_s^2}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{M}_1$$

MAGNETIZATION

Nonlinear magnetization in the magnetostrictive material is found from the following nonlinear implicit relation (Ref. 4):

$$\mathbf{M} = M_s L(|\mathbf{H}_{\mathrm{eff}}|)\frac{\mathbf{H}_{\mathrm{eff}}}{|\mathbf{H}_{\mathrm{eff}}|}$$

where L is the Langevin function

$$L = \coth\left(\frac{3\chi_m|\mathbf{H}_{\mathrm{eff}}|}{M_s}\right) - \frac{M_s}{3\chi_m|\mathbf{H}_{\mathrm{eff}}|}$$

with χ_m being the magnetic susceptibility in the initial linear region.

Other possible choices of the L function are a hyperbolic tangent

$$L = \tanh\left(\frac{\chi_m|\mathbf{H}_{\mathrm{eff}}|}{M_s}\right)$$

and a linear function

$$L = \frac{\chi_m|\mathbf{H}_{\mathrm{eff}}|}{M_s}$$

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

$$\mathbf{H}_{\mathrm{eff}} = \mathbf{H} + \frac{3}{\mu_0 M_s^2}\left[\lambda_{100}\mathrm{S}_{\mathrm{ed}} + (\lambda_{111} - \lambda_{100})\sum_{i \neq j}(S_{\mathrm{ed}})_{ij}(\mathbf{e}_i \otimes \mathbf{e}_j)\right]\mathbf{M} \qquad (3\text{-}49)$$

where **H** is the applied magnetic field. The second term in Equation 3-49 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

$$S_{\mathrm{ed}} = \mathrm{dev}(\mathrm{c_H}\varepsilon)$$

For isotropic materials, the effective magnetic field expression simplifies into

$$\mathbf{H}_{\mathrm{eff}} = \mathbf{H} + \frac{3\lambda_s}{\mu_0 M_s^2}S_{\mathrm{ed}}\mathbf{M}$$

In addition, the magnetization and magnetic field are related to each other and to the magnetic flux density (also called the **B**-field) by

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$$

COMSOL Multiphysics solves for the magnetic vector potential **A** whose curl yields the vector **B**-field. The **H**-field is then obtained as a function of the **B**-field and magnetization.

Rigid Domain Material 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, 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 $\mathbf{R}$ 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 $\mathbf{R}$ is:

$$R = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For the geometrically linear case, the 2D rotation matrix is reduced to:

$$R = \begin{bmatrix} 1 & -\phi & 0 \\ \phi & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Under translation and rotation of a rigid domain, the complete expression for the displacement of any point on the rigid body is given by:

$$\mathbf{u}_\mathrm{d} = \mathbf{u} + (R - I) \cdot (\mathbf{X} - \mathbf{X}_\mathrm{M})$$

where $\mathbf{X}$ are the material coordinates of any point in the rigid domain, $\mathbf{X}_\mathrm{M}$ is the center of mass of the rigid domain, $\mathbf{u}$ is the translation vector at the center of mass, and I is the identity matrix.

The rigid body displacement at the center of mass ($\mathbf{u}$) are degrees of freedom. Thus the rigid body translational velocity and acceleration can be evaluated by directly taking the time derivatives of $\mathbf{u}$. In the time domain it can be expressed as:

$$\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t} \qquad \ddot{\mathbf{u}} = \frac{\partial}{\partial t}\dot{\mathbf{u}}$$

In the frequency domain, they can be expressed in terms of frequency (ω):

$$\dot{\mathbf{u}} = (i\omega)\mathbf{u} \qquad \ddot{\mathbf{u}} = (i\omega)\dot{\mathbf{u}} = -\omega^2\mathbf{u}$$

The same is true for the rotation in 2D since the rigid body rotation ϕ is the degree of freedom. The rigid body angular velocity and acceleration can be evaluated by directly taking the time derivatives of ϕ.

In 3D, the situation is different and the total rotation of the rigid domain can be presented as a function of quaternion:

$$\Theta = \frac{2\,\mathrm{acos}(a)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

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:

$$\dot{\Theta} = \begin{bmatrix} Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \qquad Q = 2(\bar{q} \otimes \dot{q}) \qquad q = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Here $\bar{q}$ is the conjugate of q, and the symbol $\otimes$ denotes quaternion multiplication.

The angular acceleration of the rigid domain can be evaluated by taking the time derivative of the angular velocity.

RIGID DOMAIN DYNAMICS

The governing equation for a rigid domain can be written as a balance between the inertial (internal) forces and applied external forces. A rigid domain has only one internal force, the inertial force. This means that only the mass density of a domain is required to define the rigid domain material model.

The inertial forces and inertial moments about the center of mass are:

$$\mathbf{F} = m\ddot{\mathbf{u}} \qquad \mathbf{M} = I\ddot{\Theta}$$

where $\ddot{\mathbf{u}}$ and $\ddot{\Theta}$ are the linear and angular accelerations of a rigid domain.

The inertial properties mass (m) and moment of inertia tensor (I) of a rigid domain are computed as:

$$m = \int \rho dV$$

$$\mathbf{X}_{\mathrm{M}} = \frac{\int \rho \mathbf{X} dV}{m}$$

$$I = \int (((\mathbf{X} - \mathbf{X}_{\mathrm{M}})^T \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}}))E_3 - (\mathbf{X} - \mathbf{X}_{\mathrm{M}}) \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}})^T)\rho dV$$

where E_3 and $\mathbf{X}_{\mathrm{M}}$ are the identity matrix and the center of mass of a rigid domain, respectively. The special case for the Shell interface is described in Rigid Domain for Shells.

In 2D, the expressions for inertial forces, inertial moments, and moment of inertia reduce to:

$$\mathbf{F} = m\ddot{\mathbf{u}} \qquad M = I_z\ddot{\phi}$$

$$I_z = \int((\mathbf{X} - \mathbf{X}_\mathrm{M}) \cdot (\mathbf{X} - \mathbf{X}_\mathrm{M}))\rho h\, dA$$

where the volume integration has been replaced by an area integration multiplied by the out-of-plane thickness h.

The equations of motion for the rigid domain are:

$$m\ddot{\mathbf{u}} + \sum \mathbf{F}_\mathrm{I} = \sum \mathbf{F}_\mathrm{ext}$$

and

$$RIR^T\ddot{\Theta} + \dot{\Theta} \times (RIR^T\dot{\Theta}) + \sum \mathbf{M}_\mathrm{I} = \sum \mathbf{M}_\mathrm{ext}$$

Here, the subscripts 'I' and 'ext' 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

$$I_z\ddot{\varphi} + \sum M_\mathrm{I} = \sum M_\mathrm{ext}$$

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 the 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 ($\boldsymbol{\omega}$) about a center of rotation ($\mathbf{X}_c$), the rigid body displacement and quaternion degrees of freedom are initialized as:

$$\mathbf{u}_0 = \mathbf{u} + \mathbf{u}_\mathrm{r}$$

$$\mathbf{u}_\mathrm{r} = (-\mathbf{r} + \hat{\boldsymbol{\Omega}}(\hat{\boldsymbol{\Omega}} \cdot \mathbf{r}))(1 - \cos\phi) + (\hat{\boldsymbol{\Omega}} \times \mathbf{r})\sin\phi$$

$$\mathbf{r} = \mathbf{X}_\mathrm{M} - \mathbf{X}_\mathrm{c}$$

$$\left(\frac{\partial\mathbf{u}}{\partial t}\right)_0 = (\boldsymbol{\omega} \times (\mathbf{r} + \mathbf{u}_\mathrm{r})) + \frac{\partial\mathbf{u}}{\partial t}$$

$$\begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0 = \begin{bmatrix} \cos\frac{\phi}{2} \\ \sin\frac{\phi}{2}\hat{\boldsymbol{\Omega}} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial a}{\partial t} \\ \frac{\partial \mathbf{b}}{\partial t} \end{bmatrix}_0 = \frac{1}{2}\left(\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix}_0\right)$$

In 2D, the expressions for the initial values reduce to:

$$\mathbf{u}_0 = \mathbf{u} + \mathbf{u}_\mathrm{r}$$

$$\mathbf{u}_\mathrm{r} = -\mathbf{r}(1 - \cos\phi) + (\mathbf{e}_z \times \mathbf{r})\sin\phi$$

$$\mathbf{r} = \mathbf{X}_\mathrm{M} - \mathbf{X}_\mathrm{c}$$

$$\left(\frac{\partial\mathbf{u}}{\partial t}\right)_0 = (\mathbf{e}_z \times (\mathbf{r} + \mathbf{u}_\mathrm{r}))\frac{\partial\phi}{\partial t} + \frac{\partial\mathbf{u}}{\partial t}$$

$$\phi_0 = \phi$$

$$\left(\frac{\partial\phi}{\partial t}\right)_0 = \frac{\partial\phi}{\partial t}$$

The variable $\mathbf{u}_\mathrm{r}$ is the translation at the center of mass due to a rotation around the center of rotation, and is thus zero when the two points coincide. In the case that you are entering the data using a separate center of rotation, you must pay special attention to how the initial displacement and velocity are composed if initial rotations and rotational velocities are present.

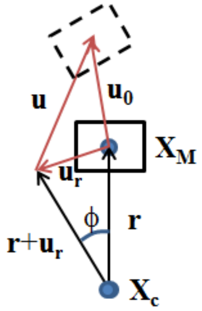


Figure 3-16: Initial displacement of a rigid body

MASS AND MOMENT OF INERTIA

Sometimes a rigid domain needs the added effect of an associated abstract rigid object, which is physically not modeled and where the inertial properties are known. You can model this using Mass and Moment of Inertia, where the inertial properties of this abstract domain (center of mass, mass, and moment of inertia tensor) can be directly entered.

The formulation for an abstract rigid object is similar to the physical rigid domain with these exceptions:

- The inertial properties of the abstract rigid object are input by the user instead of being computed from a physical domain.
- No extra degrees of freedom are created. The inertial forces generated by this feature will be computed based on the distance from the center of gravity of the rigid domain to which it belongs, and the values of the degrees of freedom there.

The inertial force contributions are

$$\mathbf{F}_{\mathrm{I}} = m\frac{d^2}{dt^2}(\mathbf{u} + (R - E_3)\mathbf{X}_{\mathrm{mc}})$$

where $\mathbf{X}_{\mathrm{mc}}$ is the vector from the center of mass of the rigid domain ($\mathbf{X}_{\mathrm{M}}$) to the center of mass of this contribution ($\mathbf{X}_{\mathrm{m}}$),

$$\mathbf{X}_{\mathrm{mc}} = \mathbf{X}_{\mathrm{m}} - \mathbf{X}_{\mathrm{M}}$$

The inertial moment contributions are

$$\mathbf{M}_{\mathrm{I}} = RIR^T\ddot{\Theta} + \dot{\Theta}\times(RIR^T\dot{\Theta}) + \mathbf{X}_{\mathrm{mc}}\times\mathbf{F}_{\mathrm{I}}$$

In 2D, there is only a scalar moment contribution:

$$M_{\mathrm{I}} = I_z\ddot{\varphi} + (\mathbf{X}_{\mathrm{mc}}\times\mathbf{F}_{\mathrm{I}})\cdot\mathbf{e}_z$$

CONSTRAINTS

The constraints for a rigid domain are different in nature than those applied to flexible 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 non-zero rotation around an arbitrary axis.

The displacement and rotation can be prescribed in a selected coordinate system about an arbitrary center of rotation. The center of rotation can be defined using

- The center of mass of the rigid domain.
- Global coordinates of the center of rotation.
- The centroid of a set of selected entities (boundaries, edges, or points).

The displacement at the center of rotation is computed as:

$$\mathbf{u}_{\mathrm{c}} = \mathbf{u} + (\mathbf{R}-\mathbf{I})\cdot(\mathbf{X}_{\mathrm{c}}-\mathbf{X}_{\mathrm{M}}) \qquad (3\text{-}50)$$

The components of this displacement vector are prescribed individually in the selected coordinate system. Through Equation 3-50, 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:

$$\mathbf{b} = \hat{\mathbf{\Omega}} \sin\frac{\phi_0}{2}$$

where $\hat{\mathbf{\Omega}}$ and ϕ_0 are the axis of rotation and angle of rotation respectively.

In 2D, the out-of-plane rotation angle is directly constrained to the prescribed value of the rotation.

LOADS

The loads available for a flexible domain can also be used for a rigid domain. In addition to these boundary conditions, a rigid domain also has global subnodes for applying forces and moments. If you use **Applied Force**, a force and its location can be prescribed in a selected coordinate system. A force implicitly also contributes to the moment unless it is applied at the center of mass of a rigid domain. If an **Applied Moment** node is used, a moment can be prescribed in a selected coordinate system.

CONNECTING TO OTHER BODIES

When a rigid domain and a flexible domain share a boundary (Shell: edge, Beam: point), the connection is automatic. All displacements on the flexible domain are controlled by the degrees of freedom of the rigid domain, so that

$$\mathbf{u}_{\mathrm{flex}} = (\mathbf{R} - \mathbf{I}) \cdot (\mathbf{X} - \mathbf{X}_{\mathrm{M}}) + \mathbf{u}$$

where **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.

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 grater 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/Pa^2) and f is a second rank tensor (SI unit: 1/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 indices)

$$F_{ii} > 0 \text{ and } F_{ii}F_{jj} \geq F_{ij}^2$$

The *failure index* is computed from the failure criterion as

$$fi = g(\sigma) + 1$$

so failure is predicted for a failure index greater than one, $fi \geq 1$.

The *damage index* is given by a boolean expression based on the failure criterion

$$di = \begin{matrix} 1 & g(\sigma) \geq 0 \\ 0 & \text{otherwise} \end{matrix}$$

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

$$g(s_{\mathrm{f}}\sigma) = 0$$

For a quadratic failure criterion, as the Tsai-Wu criterion, this means solving a quadratic equation for the safety factor variable s_{f}

$$s_{\mathrm{f}}^2(\sigma:(\boldsymbol{F}\sigma)) + s_{\mathrm{f}}(f:\sigma) - 1 = 0$$

the safety factor is then obtained from the smallest positive root.

For a isotropic criteria, such as von Mises criterion, $g(\sigma) = \sigma_{\mathrm{mises}}/\sigma_{\mathrm{ts}}-1$, and the safety factor is given by $s_{\mathrm{f}} = \sigma_{\mathrm{ts}}/\sigma_{\mathrm{mises}}$.

The *margin of safety* (Ref. 23) is then computed from the safety factor

$$ms = s_{\mathrm{f}} - 1$$

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 Material Models.

Following Tsai-Wu formalism, different orthotropic criteria can be defined by setting appropriate values for the coefficients in $\boldsymbol{F}$ and f tensors.

ANISOTROPIC TSAI-WU CRITERION

For this criterion, enter twenty one coefficients to define the 6-by-6 matrix $\boldsymbol{F}$, 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

$$F_{11} = \frac{1}{\sigma_{cs1}\sigma_{ts1}}\ ,\ F_{22} = \frac{1}{\sigma_{cs2}\sigma_{ts2}}\ ,\ F_{33} = \frac{1}{\sigma_{cs3}\sigma_{ts3}}\ ,$$

$$F_{44} = \frac{1}{\sigma^2_{ss23}}\ ,\ F_{55} = \frac{1}{\sigma^2_{ss13}}\ ,\ F_{66} = \frac{1}{\sigma^2_{ss12}}\ ,$$

$$F_{12} = -\frac{1}{2}(F_{11}+F_{22}-F_{33})\ ,\ F_{13} = -\frac{1}{2}(F_{11}+F_{33}-F_{22})\ ,$$

$$F_{23} = -\frac{1}{2}(F_{22}+F_{33}-F_{11})$$

$$f_1 = \frac{1}{\sigma_{ts1}} - \frac{1}{\sigma_{cs1}}\ ,\ f_2 = \frac{1}{\sigma_{ts2}} - \frac{1}{\sigma_{cs2}}\ ,\ f_3 = \frac{1}{\sigma_{ts3}} - \frac{1}{\sigma_{cs3}}$$

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

$$g(\varepsilon) = \max\left(\frac{\varepsilon_1}{\varepsilon_{s1}}, \frac{\varepsilon_2}{\varepsilon_{s2}}, \frac{\varepsilon_3}{\varepsilon_{s3}}, \frac{\gamma_{23}}{\gamma_{ss23}}, \frac{\gamma_{13}}{\gamma_{ss13}}, \frac{\gamma_{12}}{\gamma_{ss12}}\right) - 1$$

here, ε_{si} is either the ultimate tensile strain or the ultimate compressive strain depending whether the strain in the i-direction, ε_i, is positive or negative. The absolute value of the shear strain γ_{ij} in the ij-plane is compared to the corresponding ultimate shear strain γ_{ssij}.

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

$$F_{11} = \frac{1}{\sigma_{ts1}^2} \qquad \text{for } \sigma_{11} > 0\text{, or } F_{11} = \frac{1}{\sigma_{cs1}^2} \qquad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{ts2}^2} \qquad \text{for } \sigma_{22} > 0\text{, or } F_{22} = \frac{1}{\sigma_{cs2}^2} \qquad \text{for } \sigma_{22} < 0$$

$$F_{12} = -\frac{1}{2}\frac{1}{\sigma_{ts1}^2} \qquad \text{for } \sigma_{11}\sigma_{22} > 0 \text{ or } F_{12} = -\frac{1}{2}\frac{1}{\sigma_{cs1}^2} \qquad \text{for } \sigma_{11}\sigma_{22} < 0$$

$$F_{66} = \frac{1}{\sigma_{ss12}^2}$$

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 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

$$F_{11} = \frac{1}{\sigma_{\mathrm{ts1}}^2} \qquad \text{for } \sigma_{11} > 0\text{, or } F_{11} = \frac{1}{\sigma_{\mathrm{cs1}}^2} \qquad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{\mathrm{ts2}}^2} \qquad \text{for } \sigma_{22} > 0\text{, or } F_{22} = \frac{1}{\sigma_{\mathrm{cs2}}^2} \qquad \text{for } \sigma_{22} < 0$$

$$F_{12} = -\frac{1}{2}\frac{1}{\sigma_{\mathrm{ts1}}^2} \qquad \text{for } \sigma_{11} > 0 \text{ or } F_{12} = -\frac{1}{2}\frac{1}{\sigma_{\mathrm{cs1}}^2} \qquad \text{for } \sigma_{11} < 0$$

$$F_{66} = \frac{1}{\sigma_{\mathrm{ss12}}^2}$$

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 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

$$F_{11} = \frac{1}{\sigma_{\mathrm{ts1}}^2} \qquad \text{for } \sigma_{11} > 0\text{, or } F_{11} = \frac{1}{\sigma_{\mathrm{cs1}}^2} \qquad \text{for } \sigma_{11} < 0$$

$$F_{22} = \frac{1}{\sigma_{\mathrm{ts2}}^2} \qquad \text{for } \sigma_{22} > 0 \text{, or } F_{22} = \frac{1}{\sigma_{\mathrm{cs2}}^2} \qquad \text{for } \sigma_{22} < 0$$

$$F_{66} = \frac{1}{\sigma_{\mathrm{ss12}}^2}$$

$$F_{12} = -\frac{1}{2}\sqrt{F_{11}F_{22}}$$

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}

$$g(\sigma) = \frac{\sigma_{\mathrm{mises}}}{\sigma_{\mathrm{ts}}} - 1$$

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}

$$g(\sigma) = \frac{\sigma_{\mathrm{tresca}}}{\sigma_{\mathrm{ts}}} - 1$$

here, Tresca effective stress is defined in terms of principal stresses, $\sigma_{\mathrm{tresca}} = \sigma_1 - \sigma_3$, 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

St. Venant criterion is similar to 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

$$g(\varepsilon) = \max\left(\frac{\varepsilon_{p1}}{\varepsilon_s}, \frac{\varepsilon_{p2}}{\varepsilon_s}, \frac{\varepsilon_{p3}}{\varepsilon_s}\right) - 1$$

here, ε_s is either the ultimate tensile strain or the ultimate compressive strain depending 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.

ISOTROPIC MOHR-COULOMB CRITERION

Mohr-Coulomb criterion is similar to 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

$$g(\sigma) = \frac{1}{2}\left(\frac{\sigma_{p1} - \sigma_{p3}}{c\cos(\phi)}\right) + \frac{1}{2}\left(\frac{\sigma_{p1} + \sigma_{p3}}{c\cos(\phi)}\right)\sin(\phi) - 1$$

here, the principal stresses are sorted as $\sigma_{p1} \ge \sigma_{p2} \ge \sigma_{p3}$.

The cohesion and the angle of internal friction are related to the tensile and compressive strengths by the expressions

$$\sin(\phi) = \frac{\sigma_{cs1} - \sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}} \quad \text{and} \quad c\cos(\phi) = \frac{\sigma_{cs1}\sigma_{ts1}}{\sigma_{cs1} + \sigma_{ts1}}$$

ISOTROPIC DRUCKER-PRAGER CRITERION

Drucker-Prager criterion approximates the Mohr-Coulomb criterion by a smooth function (a cone in the stress space). The failure criterion is computed from the stress invariants I_1 and J_2, and two material parameters, α and k, see Drucker-Prager Criterion for soil plasticity

$$g(\sigma) = \frac{\sqrt{J_2}}{k - \alpha I_1} - 1$$

The material parameters α and k are related to the cohesion c and angle of internal friction ϕ in the Mohr-Coulomb criterion by the expressions

$$\alpha = \frac{2}{\sqrt{3}} \cdot \frac{\sin(\phi)}{3 \pm \sin(\phi)} \text{ and } k = \frac{2\sqrt{3}c\cos(\phi)}{3 \pm \sin(\phi)}$$

The symbol ± is related to either matching the tensile meridian (positive sign) or matching the compressive meridian (negative sign) of Mohr-Coulomb's pyramid. Also, the cohesion and the angle of internal friction are related to the tensile and compressive strengths, see Isotropic Mohr-Coulomb criterion.

USER DEFINED

The user defined 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

In this section:

- Equation Implementation

Equation Implementation

The equilibrium equations for solid mechanics are given by Newton's second law. It is usually written using a spatial formulation in terms of the Cauchy stress tensor σ:

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_x\sigma + \mathbf{f}_\mathrm{V}$$

Here $\mathbf{f}_\mathrm{V}$ 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:

$$\rho_0\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_X P + \mathbf{F}_\mathrm{V}$$

Now that the first Piola-Kirchhoff stress tensor, P, is used. $\mathbf{F}_\mathrm{V}$ is a body force with components in the current configuration but given with respect to the undeformed volume, and ρ_0 is the initial mass density. Note the gradient operators are not the same: in the first case the gradient is taken with respect to the spatial coordinates, and in the second case with respect to the material coordinates. Using the more common second Piola-Kirchhoff stress tensor, S, the same equation reads

$$\rho_0\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla_X(FS) + \mathbf{F}_\mathrm{V} \qquad (3\text{-}51)$$

where F is the deformation gradient. The COMSOL Multiphysics implementation of the equations in the Solid Mechanics interface is however not based on the equation of motion directly, but rather on the *principle of virtual work*.

The principle of virtual work states that the sum of the internal virtual work and the external virtual work are equal. The internal virtual work is the work done by the current stress state on a kinematically admissible variation in strains. The external virtual work is the work done by all forces (acting on domains, boundaries, edges, or points) when multiplied with the variation in displacements corresponding to the variation in strains. The virtual displacements $\delta\mathbf{u}$ are in the finite element formulation represented by the `test()` operator in COMSOL Multiphysics. For a stationary case, the virtual work δW is written as

$$\delta W = \int_V (-\delta\varepsilon : \sigma + \delta\mathbf{u}\cdot\mathbf{F}_V)dv + \int_S (\delta\mathbf{u}\cdot\mathbf{F}_S)ds + \int_L (\delta\mathbf{u}\cdot\mathbf{F}_L)dl + \sum_p (\delta\mathbf{u}\cdot\mathbf{F}_p)]$$

The strains are computed from the gradients of the displacements, and the stresses are given by the constitutive relation.

In a dynamic analysis, the inertial forces are included in the volume forces, according to d'Alembert's principle.

$$\delta W = \int_V (-\delta\varepsilon : \sigma + \delta\mathbf{u}\cdot\mathbf{F}_V - \rho\delta\mathbf{u}\cdot\mathbf{u}_{tt})dv + \int_S (\delta\mathbf{u}\cdot\mathbf{F}_S)ds + \int_L (\delta\mathbf{u}\cdot\mathbf{F}_L)dl + \sum_p (\delta\mathbf{u}\cdot\mathbf{F}_p) \tag{3-52}$$

Since the equations are formulated on the material frame, all integrals are taken over the undeformed geometry. The stress and strain contributions must be interpreted differently depending on whether the formulation is geometrically nonlinear or not.

- If the study step is geometrically linear, the strain ε is the engineering strain. The stress σ could in principle be any of the stress measures, as they all converge to the same engineering stress in this case.
- If the study step is geometrically nonlinear, the strain ε is the Green-Lagrange strain and the stress σ 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*.

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

$$F_{m,\,\mathrm{freq}} = F_m(f)\cos(\omega t + \phi_m)$$

$$\mathbf{F}_{\mathrm{freq}} = \begin{bmatrix} F_{x,\,\mathrm{freq}} \\ F_{y,\,\mathrm{freq}} \\ F_{z,\,\mathrm{freq}} \end{bmatrix}$$

Assume a harmonic response with the same angular frequency as the excitation load

$$u = u_{\mathrm{amp}}\cos(\omega t + \phi_u)$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The relationship can also be described using complex notation with

$$u = \mathrm{Re}(u_{\mathrm{amp}}e^{j\phi_u}e^{j\omega t}) = \mathrm{Re}(\tilde{u}e^{j\omega t}) \text{ where } \tilde{u} = u_{\mathrm{amp}}e^{j\phi_u}$$

$$\mathbf{u} = \mathrm{Re}(\tilde{\mathbf{u}}e^{j\omega t})$$

and

$$F_{m,\,\mathrm{freq}} = \mathrm{Re}(F_m(\omega)e^{j\phi_m}e^{j\omega t}) = \mathrm{Re}(\tilde{F}_m e^{j\omega t})$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{bmatrix}$$

The primary results, such as displacements, velocities, accelerations, and stress and strain components are all complex valued.

EIGENFREQUENCY STUDIES

The eigenfrequency equations are derived by assuming a harmonic displacement field, similar as for the frequency response formulation. The difference is that this study type uses a new variable $j\omega$ explicitly expressed in the eigenvalue $j\omega = -\lambda$. The eigenfrequency f is then derived from $j\omega$ as

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

Damped eigenfrequencies can also be studied, so λ is not necessarily a purely imaginary number. Any damping included in the problem will automatically cause the eigenfrequencies to become complex valued.

In addition to the eigenfrequency, the quality factor, Q, and decay factor, δ, for the model can be examined:

$$Q = \frac{\mathrm{Im}(\lambda)}{2\mathrm{Re}(\lambda)}$$

$$\delta = \mathrm{Re}(\lambda)$$

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

$$\delta W = \int_V (-\delta\varepsilon : \sigma - \lambda\delta(\varepsilon_{\mathrm{GL}} - \varepsilon) : \sigma_1) dv = 0 \tag{3-53}$$

Here ε us the engineering strain, $\varepsilon_{\mathrm{GL}}$ is the Green-Lagrange strain and σ_1 is the stress caused by the unit load. In terms of stiffness matrices, this corresponds to

$$(K_{\mathrm{L}} + \lambda K_{\mathrm{NL}}(\mathbf{u}_0))\mathbf{u} = 0$$

where K_{L} is the linear stiffness matrix, and K_{NL} is the nonlinear contribution to the full stiffness matrix. The symbolic linearization point $\mathbf{u}_0$ is the displacement vector caused by the unit load.

Strictly speaking, this formulation assumes that geometric nonlinearity is not used in the eigenvalue step. The Green-Lagrange tensor is inserted explicitly in the second term of Equation 3-53, while the first term uses the linear (engineering) strain tensor.

If, however, geometric nonlinearity is selected, then Equation 3-53 is replaced by

$$\delta W = \int_V (-\delta\varepsilon_{\mathrm{GL}} : \sigma - (\lambda - 1)\delta(\varepsilon_{\mathrm{GL}} - \varepsilon) : \sigma_1) dv = 0$$

By using the term (λ-1), the effect of using the Green-Lagrange strain tensor in the first term is to a large extent removed. Unless the unit load is significantly larger than the buckling load, the result will be the same as the intended, even if geometric nonlinearity was inadvertently selected in the eigenvalue study step.

Damping

For dynamic problems, the damping of the structure is usually an important property.

In this section:

- Rayleigh Damping
- Loss Factor Damping
- Viscous Damping

Rayleigh Damping

Rayleigh damping is described by two coefficients: the mass damping coefficient α_{dM} and the stiffness damping coefficient β_{dK}. Rayleigh damping will give the following contribution to the virtual work

$$\delta W = \int_V \left(-\delta(\nabla \mathbf{u}) : \beta_{\mathrm{dK}}\frac{\partial P}{\partial t} - \alpha_{\mathrm{dM}}\rho\delta\mathbf{u}\cdot\frac{\partial \mathbf{u}}{\partial t}\right)dv$$

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

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

$$D^\mathrm{c} = (1 + j\eta_\mathrm{s})D$$

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 $(1 + j\eta_\mathrm{s})$. 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

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

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:

$$
\begin{aligned}
E_m^{\mathrm{c}} &= (1+j\eta_{\mathrm{E},m})E_m \\
G_m^{\mathrm{c}} &= (1+j\eta_{\mathrm{G},m})G_m \\
m &= 1, 2, 3
\end{aligned}
$$

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:

$$S = (1+j\eta_{\mathrm{s}})\frac{\partial W_s}{\partial E}$$

For loss factor damping, the following definition is used for the elastic part of the entropy:

$$S_{\mathrm{elast}} = \alpha : (s - j\eta_{\mathrm{s}}(\mathrm{C} : \varepsilon))$$

note that S_{elast} here denotes the entropy contribution and not any stress.

This is because the entropy is a function of state and thus independent of the strain rate, while the damping represents the rate-dependent effects in the material (for example, viscous or viscoelastic effects). The internal work of such inelastic forces averaged over the time period $2\pi/\omega$ can be computed as:

$$Q_{\mathrm{h}} = \frac{1}{2}\omega\eta_{\mathrm{s}}\mathrm{Real}(\varepsilon : \mathrm{Conj}(\mathrm{C} : \varepsilon))$$

Q_{h} can be used as a heat source for modeling of the heat generation in vibrating structures, when coupled with the frequency-domain analysis for the stresses and strains.

Viscous Damping

Viscous damping can be added to the material models. It will cause an extra stress S_{q} proportional to the rate of elastic strain in the material,

$$S_{\mathrm{q}} = \eta_{\mathrm{b}}\dot{\varepsilon}_{\mathrm{el,\,vol}} + \eta_{\mathrm{v}}\dot{\varepsilon}_{\mathrm{el,\,dev}}$$

where η_{b} and η_{v} are the bulk and shear viscosity coefficients, respectively.

Viscous damping can be used in both frequency and time domain analyses.

In case of geometric nonlinearity, the viscous stress is treated as being a Cauchy stress acting in the actual configuration (spatial frame). The resulting contribution to the second Piola-Kirchhoff stress is calculated as

$$S_\mathrm{q} = \left(\eta_\mathrm{b} - \frac{2}{3}\eta_\mathrm{v}\right)\dot{J}_\mathrm{el}C_\mathrm{el}^{-1} + \eta_\mathrm{v}C_\mathrm{el}^{-T}\dot{\varepsilon}_\mathrm{el}C_\mathrm{el}^{-1}$$

where 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

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 **f** acting on the current area da and the specified distributed load **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

$$\mathbf{F} = p\mathbf{n}\frac{da}{dA}\left(1 + \frac{\partial w}{\partial z}\right)$$

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:

$$\mathbf{a}_{\mathrm{tot}} = \ddot{\mathbf{u}} + \mathbf{a}_{\mathbf{f}}$$

$$\mathbf{a}_{\mathbf{f}} = \mathbf{a}_{\mathrm{cen}} + \mathbf{a}_{\mathrm{cor}} + \mathbf{a}_{\mathrm{eul}}$$

GRAVITY

The gravity acts in a fixed spatial direction $\mathbf{e}_{\mathrm{g}}$. The intensity is

$$\mathbf{g} = \rho g \mathbf{e}_{\mathrm{g}}$$

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 outwards from the axis of rotation defined by the axial direction vector $\mathbf{e}_{\mathrm{ax}}$. The rotation is represented by the angular velocity vector:

$$\boldsymbol{\Omega} = \Omega \mathbf{e}_{\mathrm{ax}}$$

where Ω is the angular velocity. In vector form, the acceleration contribution and the loads are:

$$\mathbf{a}_{\mathrm{cen}} = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_{\mathrm{p}})$$

$$\mathbf{F}_{\mathrm{cen}} = -\rho \mathbf{a}_{\mathrm{cen}}$$

where $\mathbf{r}_{\mathrm{p}}$ is the rotation position vector that contains the coordinates with respect to any point on the axis of rotation. The point is given by its radius vector in the global coordinate system $\mathbf{r}_{\mathrm{bp}}$.

SPIN-SOFTENING EFFECT

The structural displacement can be accounted for when computing the rotation position, so that

$$\mathbf{r}_{\mathrm{p}} = \mathbf{X} + \mathbf{u} - \mathbf{r}_{\mathrm{bp}}$$

This results in a contribution from the extra acceleration terms caused by the deformation into the system's stiffness matrix. The effect is often called spin-softening.

CORIOLIS FORCE

For a Coriolis force to appear, the object studied must have a velocity relative to the rotating frame. The acceleration contribution and the load are:

$$\mathbf{a}_{\mathrm{cor}} = 2\boldsymbol{\Omega}\times\frac{\partial\mathbf{u}}{\partial t}$$

$$\mathbf{F}_{\mathrm{cor}} = -\rho\mathbf{a}_{\mathrm{cor}}$$

This gives a damping contribution since it is proportional to the velocity.

EULER FORCE

The Euler force occurs when the rate of rotation is not constant in time. The force acts in the plane of rotation perpendicular to the centrifugal force. The acceleration contribution and the load are:

$$\mathbf{a}_{\mathrm{eul}} = \frac{\partial\boldsymbol{\Omega}}{\partial t}\times\mathbf{r}_{\mathrm{p}}$$

$$\mathbf{F}_{\mathrm{eul}} = -\rho\mathbf{a}_{\mathrm{eul}}$$

Spring Foundation and Thin Elastic Layer

In this section, the equations for the spring type physics nodes are developed using boundaries, but the generalizations to geometrical objects of other dimensions are obvious. Also, for cases where rotational springs are present, the relations between moments and rotations are analogous to the relations between forces and displacements described below.

SPRING FOUNDATION

A spring gives a force that depends on the displacement and acts in the opposite direction. In the case of a force that is proportional to the displacement, this is called Hooke's law. In a suitable coordinate system, a spring condition can be represented as

$$\mathbf{f}_{\mathrm{s}} = -K\cdot(\mathbf{u}-\mathbf{u}_0)$$

where $\mathbf{f}_s$ is a force/unit area, $\mathbf{u}$ is the displacement deforming the spring, and K is a stiffness matrix. $\mathbf{u}_0$ is an optional deformation offset, which describes the stress-free state of the spring.

If the spring stiffness is not constant, then it is in general easier to directly describe the force as a function of the displacement, so that

$$\mathbf{f}_\mathrm{s} = \mathbf{f}(\mathbf{u} - \mathbf{u}_0)$$

In the same way, a viscous damping can be described as a force proportional to the velocity

$$\mathbf{f}_\mathrm{v} = -D(\dot{\mathbf{u}} - \dot{\mathbf{u}}_0)$$

where D is a matrix representing the viscosity.

Structural ("loss factor") damping is only relevant for frequency domain analysis and is defined as

$$\mathbf{f}_\mathrm{l} = -i\eta K \cdot (\mathbf{u} - \mathbf{u}_0)$$

where η is the loss factor and i is the imaginary unit. It is also possible to give individual loss factors for each component in the stiffness matrix K.

If the elastic part of the spring definition is given as a force versus displacement relation, the stiffness K is taken as the stiffness at the linearization point at which the frequency response analysis is performed. Since the loss factor force is proportional to the elastic force, the equation can be written as

$$\mathbf{f}_\mathrm{l} = i\eta\mathbf{f}_s$$

The contribution to the virtual work is

$$\delta W = \int_A (\mathbf{f}_\mathrm{s} + \mathbf{f}_\mathrm{l} + \mathbf{f}_\mathrm{v}) \cdot \delta\mathbf{u} dA$$

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.

$$\mathbf{f}_\mathrm{sD} = -\mathbf{f}_\mathrm{sS} = -K(\mathbf{u}_\mathrm{D} - \mathbf{u}_\mathrm{S} - \mathbf{u}_0)$$

The uppercase indices refer to “source” and “destination”. When a force versus displacement description is used,

$$\mathbf{f}_{\mathrm{sD}} = -\mathbf{f}_{\mathrm{sS}} = \mathbf{f}(\mathbf{u}-\mathbf{u}_0)$$
$$\mathbf{u} = \mathbf{u}_{\mathrm{D}} - \mathbf{u}_{\mathrm{S}}$$

The viscous and structural damping forces have analogous properties,

$$\mathbf{f}_{\mathrm{vD}} = -\mathbf{f}_{\mathrm{vS}} = -D(\dot{\mathbf{u}}_{\mathrm{D}} - \dot{\mathbf{u}}_{\mathrm{S}} - \dot{\mathbf{u}}_0)$$

$$\mathbf{f}_{\mathrm{lD}} = -\mathbf{f}_{\mathrm{lS}} = -i\eta K(\mathbf{u}_{\mathrm{D}} - \mathbf{u}_{\mathrm{S}} - \mathbf{u}_0)$$

or

$$\mathbf{f}_{\mathrm{lD}} = i\eta\mathbf{f}_{\mathrm{sD}}$$

The virtual work expression is formulated on the destination side of the pair as

$$\delta W = \int_{A_D} (\mathbf{f}_{\mathrm{sD}} + \mathbf{f}_{\mathrm{lD}} + \mathbf{f}_{\mathrm{vD}}) \cdot \delta(\mathbf{u}_{\mathrm{D}} - \mathbf{u}_{\mathrm{S}}) dA_{\mathrm{D}}$$

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

$$\mathbf{f}_{\mathrm{su}} = -\mathbf{f}_{\mathrm{sd}} = -K(\mathbf{u}_{\mathrm{u}} - \mathbf{u}_{\mathrm{d}} - \mathbf{u}_0)$$

or

$$\mathbf{f}_{\mathrm{su}} = -\mathbf{f}_{\mathrm{sd}} = \mathbf{f}(\mathbf{u}-\mathbf{u}_0)$$
$$\mathbf{u} = \mathbf{u}_{\mathrm{u}} - \mathbf{u}_{\mathrm{d}}$$

The viscous force is

$$\mathbf{f}_{\mathrm{vu}} = -\mathbf{f}_{\mathrm{vd}} = -D(\dot{\mathbf{u}}_{\mathrm{u}} - \dot{\mathbf{u}}_{\mathrm{d}} - \dot{\mathbf{u}}_0)$$

and the structural damping force is

$$\mathbf{f}_{\mathrm{lu}} = -\mathbf{f}_{\mathrm{ld}} = -i\eta K(\mathbf{u}_{\mathrm{u}} - \mathbf{u}_{\mathrm{d}} - \mathbf{u}_0)$$

or

$$\mathbf{f}_{\mathrm{lu}} = i\eta\mathbf{f}_{\mathrm{su}}$$

The subscripts ‘u’ and ‘d’ denote the “up” and “down” sides of the interior boundary, respectively.

The virtual work expression is formulated as

$$\delta W = \int_{A_u} (\mathbf{f}_{\mathrm{sd}} + \mathbf{f}_{\mathrm{ld}} + \mathbf{f}_{\mathrm{vd}}) \cdot \delta(\mathbf{u}_{\mathrm{u}} - \mathbf{u}_{\mathrm{d}}) dA_{\mathrm{u}}$$

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

$$k_{\mathrm{n}} = \frac{E(1-\nu)}{d_{\mathrm{s}}(1+\nu)(1-\nu)}$$

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

$$k_{\mathrm{t}} = \frac{G}{d_{\mathrm{s}}} = \frac{E}{2d_{\mathrm{s}}(1+\nu)}$$

Since the layer thickness is known in this case, it is also possible to compute a strain in the elastic layer. The strain tensor has the is stored in a variable with a name like `<`*`interface`*`>.<`*`feature`*`>.etel`*`ij`*, for example `solid.tel1.etelxx` for the normal strain. The two shear strains are stored in the xy and xz components of the tensor. In 3D, the orientation of the two local directions y and z used for the two shear strain directions is obtained using the following scheme:

1. Choose an auxiliary direction. Unless the normal to the layer is very close to the global X-direction, use $\mathbf{e}_{\mathrm{aux}} = \mathbf{e}_X$. If the X-direction cannot be used, the Y-direction is instead used as the auxiliary direction, $\mathbf{e}_{\mathrm{aux}} = \mathbf{e}_Y$.
2. The local y-direction is obtained from the part of the auxiliary direction which is orthogonal to the normal direction $\mathbf{n}$:

$$\mathbf{e}_y = \frac{\mathbf{e}_{\mathrm{aux}} - (\mathbf{e}_{\mathrm{aux}} \cdot \mathbf{n})\mathbf{n}}{|\mathbf{e}_{\mathrm{aux}} - (\mathbf{e}_{\mathrm{aux}} \cdot \mathbf{n})\mathbf{n}|}$$

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

$$\mathbf{f}_{\mathrm{m}} = -M\frac{\partial^2\mathbf{u}}{\partial t^2}$$

where M is a diagonal mass distribution matrix. For added mass on a boundary (and for objects of other dimensions), the contribution to the virtual work is:

$$\delta W = \int_A \mathbf{f}_{\mathrm{m}} \cdot \delta\mathbf{u} dA$$

Rigid Motion Suppression 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 p1, p2, and p3 that are not located on a common straight line.
2. Compute the unit vectors from p1 to p2 and p3:

$$\mathbf{e}_{12} = \frac{\mathbf{X}_2 - \mathbf{X}_1}{|\mathbf{X}_2 - \mathbf{X}_1|}$$

$$\mathbf{e}_{13} = \frac{\mathbf{X}_3 - \mathbf{X}_1}{|\mathbf{X}_3 - \mathbf{X}_1|}$$

3. Compute the normal to the plane spanned by these two vectors.

$$\mathbf{e}_{\mathrm{n}1} = \frac{\mathbf{e}_{12} \times \mathbf{e}_{13}}{|\mathbf{e}_{12} \times \mathbf{e}_{13}|}$$

4. This normal is perpendicular to the line between p1 and p2. Compute a second perpendicular direction, orthogonal to $\mathbf{e}_{\mathrm{n}1}$.

$$\mathbf{e}_{n2} = \mathbf{e}_{12} \times \mathbf{e}_{n1}$$

5 Fix the first point is all three directions, so that there are no rigid body translations.

$$\mathbf{u}_{p1} = 0$$

6 Constrain the second point p2 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 p3 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.

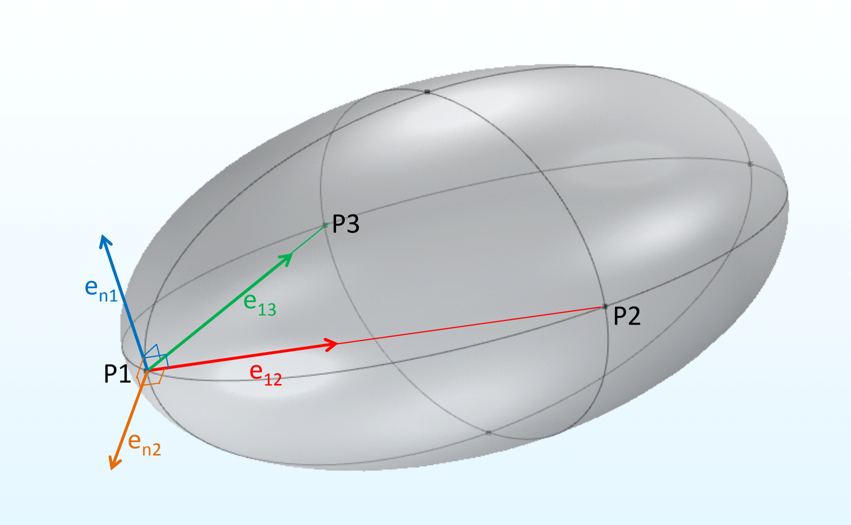


Figure 3-17: Selection of constraint orientations: P1 is fixed, P2 is constrained so that it can only move in the $\mathbf{e}_{12}$ *direction, and P3 is constrained in the* $\mathbf{e}_{n1}$ *direction.*

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:

$$\mathbf{u}_{\mathrm{p1}} = 0$$

$$\mathbf{e}_{12} = \frac{\mathbf{X}_2 - \mathbf{X}_1}{|\mathbf{X}_2 - \mathbf{X}_1|}$$

$$\mathbf{u}_{\mathrm{p2}} \cdot (\mathbf{e}_{12} \times \mathbf{e}_Z) = 0$$

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

$$\sigma \cdot \mathbf{n} = -\rho c_p \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n}\right)\mathbf{n} - \rho c_s \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{t}\right)\mathbf{t}$$

where **n** and **t** are the unit normal and tangential vectors at the boundary, respectively, and c_p and c_s are the speeds of the pressure and shear waves in the material. This approach works best when the wave direction in close to the normal at the wall.

In the general case, you can use

$$\sigma \cdot \mathbf{n} = -\mathbf{d}_i(\rho, c_p, c_s)\frac{\partial \mathbf{u}}{\partial t}$$

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 **u** is a vector of dependent variables, **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

$$\mathbf{u}_{\mathrm{destination}} = e^{-i\varphi}R_\theta\mathbf{u}^T_{\mathrm{source}}$$

where the **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

$$\mathbf{F} = -\mathbf{F}_0\exp[-im\,\mathrm{atan}(Y/X)]$$

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.

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

$$\varepsilon(\mathbf{X}) = \alpha(\mathbf{X}, T)(T - T_{\mathrm{ref}})$$

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

$$u_i(\mathbf{X}) = u_i(\mathbf{X}_0) + \int_{\mathbf{X}_0}^{\mathbf{X}}\left(\varepsilon_{il} + (X_k - X'_k)\left(\frac{\partial\varepsilon_{il}}{\partial X_k} - \frac{\partial\varepsilon_{kl}}{\partial X_i}\right)\right)dX'_l$$

Summation over double indices is implied. The rigid body rotation term given in Ref. 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 1as

$$\mathbf{X}' = \mathbf{X}_0 + s\mathbf{p}$$

giving

$$u_i(\mathbf{X}) = \int_0^1 \left(\varepsilon_{il} + (1-s)p_k\left(\frac{\partial\varepsilon_{il}}{\partial X_k} - \frac{\partial\varepsilon_{kl}}{\partial X_i}\right)\right)p_l ds \tag{3-54}$$

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 $\boldsymbol{\Theta}$ is given by

$$\boldsymbol{\Theta} = \frac{1}{2}\nabla\times\mathbf{u}$$

Applying the rotation operator to Equation 3-54 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.

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

$$\mathbf{u} = \mathbf{x} - \mathbf{X} = (R - I) \cdot (\mathbf{X} - \mathbf{X}_\mathrm{c})$$

where I is the unit matrix.

When the center of rotation of the rigid connector also has a translation $\mathbf{u}_\mathrm{c}$, then the complete expression for the rigid body displacements is

$$\mathbf{u} = (R - I) \cdot (\mathbf{X} - \mathbf{X}_\mathrm{c}) + \mathbf{u}_\mathrm{c}$$

The total rotation of the rigid connector can be also presented as a rotation vector. Its definition is

$$\Theta = \frac{2\,\mathrm{acos}(a)}{\sqrt{b^2 + c^2 + d^2}} \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

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.

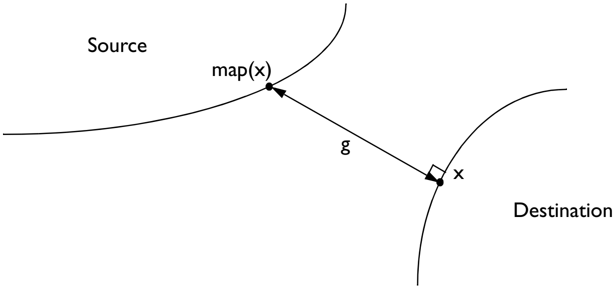
In the augmented Lagrangian method, the system of equations is solved in a segregated way. Augmentation components are introduced for the contact pressure T_n and the components $T_{\mathrm{t}i}$ of the friction traction vector $\mathbf{T}_\mathrm{t}$. An additional iteration level is added where the 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 $\boldsymbol{F}$ is the deformation gradient matrix. When looking at expressions evaluated on the destination boundaries, the expression map (E) denotes the value of the expression 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 map (E) and the gap distance variable are defined by the contact element `elcontact`. 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

$$d_{\mathrm{g}} = g - o_{\mathrm{dst}} - \frac{o_{\mathrm{src}}}{-(\mathbf{n} \cdot \mathrm{map}(\mathbf{n}))}$$

The correction of the source side offset is necessary since the normal on the source side is not necessarily pointing towards 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

$$d_{\mathrm{g}} = g - g_{\mathrm{init}} - o_{\mathrm{dst}} - \frac{o_{\mathrm{src}}}{-(\mathbf{n} \cdot \mathrm{map}(\mathbf{n}))}$$

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

$$\mathbf{s} = \mathrm{map}(F)(\mathbf{X}_{\mathrm{m}} - \mathbf{X}_{\mathrm{m,\,old}})$$

where the index 'm' indicates that the coordinates are taken as the mapped coordinates from the source side,

$$\mathbf{X}_{\mathrm{m}} = \mathrm{map}(\mathbf{X})$$

and $\mathbf{X}_{\mathrm{m,old}}$ is the value of $\mathbf{X}_{\mathrm{m}}$ 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 $\boldsymbol{F}$ 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

$$T_{\mathrm{np}} = \begin{cases} T_{\mathrm{n}}-p_{\mathrm{n}}d_{\mathrm{g}} & \text{if } \; d_{\mathrm{g}} \le 0 \\ T_{\mathrm{n}}e^{-\frac{p_{\mathrm{n}}d_{\mathrm{g}}}{T_{\mathrm{n}}}} & \text{otherwise} \end{cases} \tag{3-55}$$

p_{n} is the user-defined normal penalty factor.

The *penalized friction traction* $\mathbf{T}_{\mathrm{tp}}$ is defined on the destination boundary as:

$$\mathbf{T}_{\mathrm{tp}} = \min\left(\frac{T_{\mathrm{t,\,crit}}}{|\mathbf{T}_{\mathrm{t,\,trial}}|}, 1\right)\mathbf{T}_{\mathrm{t,\,trial}} \tag{3-56}$$

where $\mathbf{T}_{\mathrm{t,trial}}$ is defined as

$$\mathbf{T}_{\mathrm{t,\,trial}} = \mathbf{T}_{\mathrm{t}} - p_{\mathrm{t}}\mathbf{s} \tag{3-57}$$

In Equation 3-57, p_{t} is the user-defined friction traction penalty factor. The critical sliding resistance, $T_{\mathrm{t,crit}}$, is defined as

$$T_{\mathrm{t,\,crit}} = \min(\mu T_{np} + T_{\mathrm{cohe}}, T_{\mathrm{t,\,max}}) \tag{3-58}$$

In Equation 3-58, μ is the friction coefficient, T_{cohe} is the user-defined cohesion sliding resistance, and $T_{\mathrm{t,max}}$ 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:

$$\int_{\mathrm{dst}} (T_{\mathrm{np}}\delta g + \mathbf{T}_{\mathrm{tp}} \cdot \mathrm{map}(F)\delta\mathbf{X}_{\mathrm{m}})dA + \int_{\mathrm{dst}} (w_{\mathrm{cn}}\delta T_{\mathrm{n}} + \mathbf{w}_{\mathrm{ct}} \cdot \delta\mathbf{T}_{\mathrm{t}})dA$$

where $\mathbf{w}_{\mathrm{cn}}$ and $\mathbf{w}_{\mathrm{ct}}$ are contact help variables defined as:

$$\mathbf{w}_{\mathrm{cn}} = \mathbf{T}_{\mathrm{np},\,i} - \mathbf{T}_{\mathrm{n},\,i+1}$$

$$\mathbf{w}_{\mathrm{ct}} = (\gamma_{\mathrm{fric}}(\mathbf{T}_{\mathrm{tp}} - (\mathbf{n} \cdot \mathbf{T}_{\mathrm{tp}})\mathbf{n}))_i - \mathbf{T}_{\mathrm{t},\,i+1}$$

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

$$T_n = \begin{cases} -p_n d_g + p_0 & \text{if} \quad d_g < p_0 / p_n \\ 0 & \text{if} \quad d_g \geq p_0 / p_n \end{cases}$$

where d_g 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.

$$T_{t,\text{ crit}} = \begin{cases} \min(\mu T_{np} + T_{\text{cohe}}, T_{t,\text{ max}}) & \text{augmented Lagrangian method} \\ \min(\mu T_n + T_{\text{cohe}}, T_{t,\text{ max}}) & \text{penalty method} \end{cases}$$

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.

$$\int_{\text{dst}}(T_{\text{np}}\delta g + \mathbf{T}_{\text{tp}}\cdot\text{map}(F)\delta\mathbf{X}_{\text{m}})dA$$

FRICTION MODELS

The friction model is either no friction or Coulomb friction.

The frictional coefficient μ is defined as

$$\mu = \begin{cases} \mu_{\text{d}} + (\mu_{\text{s}} - \mu_{\text{d}})\exp(-\alpha_{\text{dcf}}|v_{\text{s}}|) & \text{if dynamic friction} \\ \mu_{\text{s}} & \text{otherwise} \end{cases}$$

where μ_{s} is the static frictional coefficient, μ_{d} is the dynamic frictional coefficient, v_{s} is the slip velocity, and α_{dcf} 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 the get into contact by adding an Adhesion subnode to **Contact**. Adhesion can 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 to 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

$$\sigma_{\text{n}} = -K_{\text{p}}d_{\text{g}}$$

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

$$K_\tau = n_\tau K_\mathrm{p}$$

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.

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 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 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}.

$$u_{\mathrm{m}} = \sqrt{\langle u_{\mathrm{I}}\rangle^2 + u_{\mathrm{II}}^2}$$

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. This is the displacement 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 3-18 is generic, and valid both for mode I and mode II.

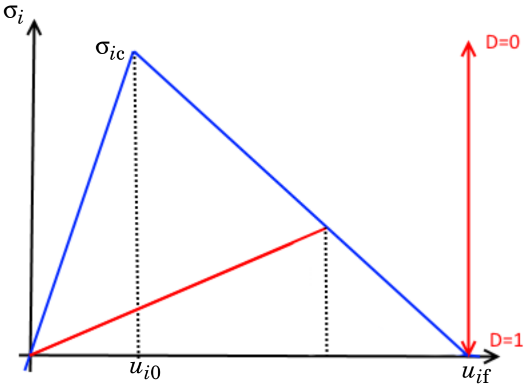


Figure 3-18: Stress vs. displacement curve for the linear separation law

The failure initiation displacement u_{i0} for each mode is determined from the maximum stress and the elastic stiffness, as

$$u_{\mathrm{I0}} = \frac{\sigma_{\mathrm{Ic}}}{K_{\mathrm{p}}}$$

$$u_{\mathrm{II0}} = \frac{\sigma_{\mathrm{IIc}}}{K_{\tau}}$$

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):

$$u_{i\mathrm{f}} = \frac{2G_{i\mathrm{c}}}{\sigma_{i\mathrm{c}}}$$

The mixed mode failure initiation displacement is then defined as

$$u_0 = u_{\mathrm{I}0}u_{\mathrm{II}0}\sqrt{\frac{u_{\mathrm{m}}^2}{\langle u_{\mathrm{I}}\rangle^2 u_{\mathrm{II}0}^2 + u_{\mathrm{II}}^2 u_{\mathrm{I}0}^2}}$$

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

$$\left(\frac{G_{\mathrm{I}}}{G_{\mathrm{I}c}}\right)^{\eta} + \left(\frac{G_{\mathrm{II}}}{G_{\mathrm{II}c}}\right)^{\eta} = 1$$

whereas the **Benzeggagh-Kenane** failure criterion is defined as

$$G_{\mathrm{I}c} + (G_{\mathrm{II}c} - G_{\mathrm{I}c})\left(\frac{G_{\mathrm{II}}}{G_{\mathrm{I}} + G_{\mathrm{II}}}\right)^{\eta} = G_{\mathrm{I}} + G_{\mathrm{II}}$$

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

$$\begin{cases} u_{\mathrm{f}} = \dfrac{2(1+\beta^2)}{u_0}\left(\left(\dfrac{K_{\mathrm{p}}}{G_{\mathrm{I}c}}\right)^{\eta} + \left(\dfrac{\beta^2 K_{\tau}}{G_{\mathrm{II}c}}\right)^{\eta}\right)^{\frac{-1}{\eta}} & u_{\mathrm{I}} > 0 \\ u_{\mathrm{f}} = u_{\mathrm{IIf}} & u_{\mathrm{I}} \le 0 \\ \beta = \dfrac{u_{\mathrm{II}}}{u_{\mathrm{I}}} & \end{cases}$$

For the Benzeggagh-Kenane criterion, the corresponding expression is

$$\begin{cases} u_{\mathrm{f}} = \dfrac{2u_{\mathrm{m}}^2}{u_0 K_{\mathrm{p}}(u_{\mathrm{I}}^2 + n_\tau u_{\mathrm{II}}^2)}\left(G_{\mathrm{Ic}} + (G_{\mathrm{IIc}} - G_{\mathrm{Ic}})\left(\dfrac{n_\tau u_{\mathrm{II}}^2}{u_{\mathrm{I}}^2 + n_\tau u_{\mathrm{II}}^2}\right)^{\eta}\right) & u_{\mathrm{I}} > 0 \\ u_{\mathrm{f}} = u_{\mathrm{IIf}} & u_{\mathrm{I}} \le 0 \end{cases}$$

The damage function can be described as

$$D(u) = \min\left(1, \max\left(0, \frac{u_{\mathrm{f}}}{u}\left(\frac{u - u_0}{u_{\mathrm{f}} - u_0}\right)\right)\right)$$

The secant stiffness of the damaged adhesive layer (indicated by the red line in Figure 3-18) is

$$K_{\mathrm{I}} = (1 - D)K_{\mathrm{p}}$$
$$K_{\mathrm{II}} = (1 - D)K_\tau$$

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.

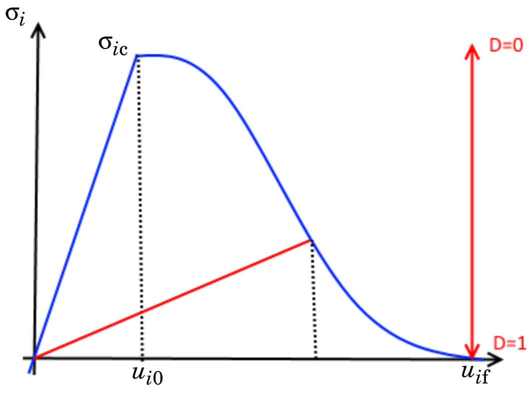


Figure 3-19: Stress vs. displacement curve for the polynomial separation law

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 3-20.

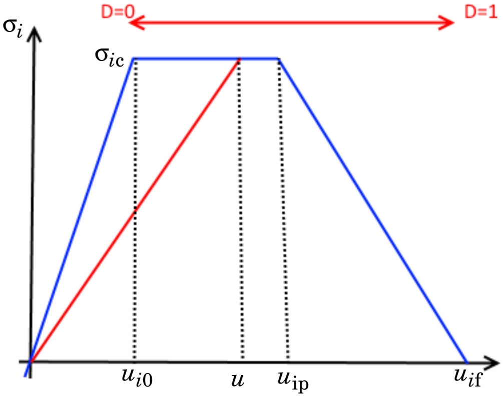


Figure 3-20: Stress vs. displacement curve for the multilinear separation law

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 λ=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

$$u_{i\mathrm{f}} = (2-\lambda)\frac{G_{i\mathrm{c}}}{\sigma_{i\mathrm{c}}} + \frac{\lambda u_{i0}}{2}$$

In this model, a linear mixed mode failure law is assumed:

$$\left(\frac{G_{\mathrm{I}}}{G_{\mathrm{I}c}}\right) + \left(\frac{G_{\mathrm{II}}}{G_{\mathrm{II}c}}\right) = 1$$

The mixed mode plastic failure displacement u_{p} is defined as

$$u_{\mathrm{p}} = u_{\mathrm{Ip}}u_{\mathrm{IIp}}\sqrt{\frac{u_{\mathrm{m}}^2}{\langle u_{\mathrm{I}}\rangle^2 u_{\mathrm{IIp}}^2 + u_{\mathrm{II}}^2 u_{\mathrm{Ip}}^2}}$$

The mixed mode failure displacement is

$$u_{\mathrm{f}} = 2G_{\mathrm{I}c}G_{\mathrm{II}c}\frac{1+\beta^2}{u_0(K_{\mathrm{p}}G_{\mathrm{II}c} + \beta^2 K_\tau G_{\mathrm{I}c})} - (u_{\mathrm{p}} - u_0)$$

The damage function can be written as

$$D(u) = \min\left(1, \max\left(0, \begin{cases} 1 - \dfrac{u_0}{u} & u < u_{\mathrm{p}} \\ 1 - \dfrac{u_0}{u}\left(\dfrac{u_{\mathrm{f}} - u}{u_{\mathrm{f}} - u_{\mathrm{p}}}\right) & u \ge u_{\mathrm{p}} \end{cases}\right)\right)$$

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 $x1$-$x2$-$x3$ where the $x1$ direction is oriented along the SCL. In 3D, you must specify the $x2$ direction, and thus implicitly the x3 direction. In 2D and 2D axial symmetry, the $x3$ direction is in the out-of-plane direction, that is the Z and azimuthal directions respectively.

The length of the SCL, which is assumed to be straight, start on one boundary, and end on the opposite boundary, is denoted L.

The *membrane stress tensor* is the average of each local stress component along the SCL:

$$\sigma_{\mathrm{m},\,ij} = \frac{1}{L}\int_0^L \sigma_{ij} dx_1$$

Each component of the *bending stress tensor* is assumed to have a linear variation along the SCL, with the value being zero at the midpoint.

$$\sigma_{\mathrm{b},\,ij} = \left(1-\frac{2x_1}{L}\right)\sigma_{\mathrm{b(max)},\,ij}$$

The maximum bending stress is defined so that the linear stress distribution has the same moment as the true stress distribution.

$$\sigma_{\mathrm{b(max)},\,ij} = \frac{6}{L^2}\int_0^L \sigma_{ij}\left(\frac{L}{2}-x_1\right)dx_1$$

The linearized stress distribution is the sum of the membrane and bending stresses,

$$\sigma_{\mathrm{mb},\,ij}(x_1) = \sigma_{\mathrm{m},\,ij}+\sigma_{\mathrm{b},\,ij}(x_1)$$

Finally the *peak stress tensor* is defined at the two end points of the SCL. It is the difference between the actual stress tensor, and the linearized stress representation.

$$\sigma_{\mathrm{p(start)},\,ij} = [\sigma_{ij} - \sigma_{\mathrm{mb},\,ij}]_{x_1 = 0} = [\sigma_{ij}]_{x_1 = 0} - \sigma_{\mathrm{m},\,ij} + \sigma_{\mathrm{b(max)},\,ij}$$
$$\sigma_{\mathrm{p(end)},\,ij} = [\sigma_{ij} - \sigma_{\mathrm{mb},\,ij}]_{x_1 = L} = [\sigma_{ij}]_{x_1 = L} - \sigma_{\mathrm{m},\,ij} - \sigma_{\mathrm{b(max)},\,ij}$$

The *stress intensity,* also known as the Tresca 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

$$\sigma_{\mathrm{int}} = \sigma_{\mathrm{p1}} - \sigma_{\mathrm{p3}}$$

When computing the principal stresses at the end points of the SCL, it is customary to ignore the bending part of the through-thickness oriented stresses. The principal stresses and the stress intensity variables are the end points are thus computed using the following stress tensor:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} \sigma_{\mathrm{m},\,11} \\ \sigma_{\mathrm{mb},\,22} \\ \sigma_{\mathrm{mb},\,33} \\ \sigma_{\mathrm{m},\,12} \\ \sigma_{\mathrm{m},\,13} \\ \sigma_{\mathrm{mb},\,23} \end{bmatrix}$$

Section forces, similar to what would be computed in a shell or plate analysis, is another type of result quantities available for each SCL. The in-plane forces are computed from the membrane stresses as

$$\begin{bmatrix} N_{22} \\ N_{33} \\ N_{23} \end{bmatrix} = L \begin{bmatrix} \sigma_{\mathrm{m},\,22} \\ \sigma_{\mathrm{m},\,33} \\ \sigma_{\mathrm{m},\,23} \end{bmatrix}$$

The bending moments are computed from the bending stresses as

$$\begin{bmatrix} M_{22} \\ M_{33} \\ M_{23} \end{bmatrix} = \frac{L^2}{6} \begin{bmatrix} \sigma_{\mathrm{b(max)},\,22} \\ \sigma_{\mathrm{b(max)},\,33} \\ \sigma_{\mathrm{b(max)},\,23} \end{bmatrix}$$

The out-of-plane shear forces are computed from the membrane stresses as

$$\begin{bmatrix} Q_2 \\ Q_3 \end{bmatrix} = L \begin{bmatrix} \sigma_{\mathrm{m},\,12} \\ \sigma_{\mathrm{m},\,13} \end{bmatrix}$$

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.

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

$$W_\mathrm{s} = \int_0^{\varepsilon_{el}} \sigma{:}d\varepsilon_\mathrm{el} \tag{3-59}$$

If the linear elasticity is assumed, then

$$\sigma = \sigma_i + C{:}\varepsilon_\mathrm{el}$$

where σ_i is the initial stress. The integration can then be carried out analytically and the result is

$$W_\mathrm{s} = \sigma_i{:}\varepsilon_\mathrm{el} + \frac{1}{2}(C{:}\varepsilon_\mathrm{el}){:}\varepsilon_\mathrm{el} = \frac{1}{2}(\sigma_i + \sigma){:}\varepsilon_\mathrm{el}$$

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 non-constant initial stress.

In the case of Frequency Domain analysis, only the harmonic part is considered. That is, a constant pre-stress does not contribute to the strain energy density. To emphasize this, the concept of *stored energy* is used. The stored energy is the cycle average of the elastic energy; that is,

$$W_h = \frac{1}{T}\int_0^T \frac{1}{2}\sigma(t):\varepsilon(t)\ dt$$

The harmonic stress and strain components are generally not in phase with each other, so the cycle average of the stored energy is computed as

$$W_h = \frac{1}{4}\mathrm{real}(\sigma:\bar{\varepsilon})$$

where the stress and strain are considered as complex quantities, and the 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 3-59 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

$$W_{\mathrm{k}} = \frac{1}{4}\rho\omega^2|v|^2$$

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

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

The treatment of dissipated energy is fundamentally different depending on whether the analysis is in frequency domain or not. For stationary or time dependent cases, the dissipated energy must be accumulated over parameter ranges or time, which means extra degrees of freedom must be added. For this reason, you must explicitly select to compute the mechanical dissipation in these cases. This is done in the **Energy Dissipation** section of the material model.

In frequency domain, the dissipated energy per cycle is computed using a closed form expression. It is always available as postprocessing variables. The expression used is

$$Q_h = \frac{1}{2}\mathrm{real}(\sigma:\overline{(i\omega\varepsilon)})$$

In the case of Rayleigh damping, the stresses are not directly affected by the damping, since it is not part of the stress-strain relation. In a problem which is linear elastic except for the Rayleigh damping, the stresses and strains are in phase with each other, and does thus not give any damping contribution. This is handled by explicitly adding an extra contribution to dissipation.

Mechanical Energy Flux

The mechanical energy flux is a vector formed by the multiplication of the stress tensor and the velocity vector

$$\mathbf{I} = -\sigma \cdot \mathbf{v}$$

The reason for the minus sign in the definition is that if you put a pressure on an 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.

$$\mathbf{I} = \frac{1}{2}\mathrm{real}(-\sigma \cdot \tilde{\mathbf{v}})$$

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[1]	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

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE[1]	COMMENT
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	
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

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE[1]	COMMENT
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
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

TABLE 3-6: ENERGY VARIABLES USED IN STRUCTURAL MECHANICS

VARIABLE	DESCRIPTION	SI UNIT	PHYSICS INTERFACE[1]	COMMENT
phys.Wmd_tot	Total damping dissipation	J	solid	Global variable containing integration and summation of all phys.Wmd contributions in a physics interface.
phys.I*	Mechanical energy flux, * coordinate	W/m^2	solid, mbd	
phys.Icomplex*	Complex mechanical energy flux, *coordinate	W/m^2	solid, mbd	Frequency domain only

[1] The availability of physics interfaces is based on the license for a module.

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. M. Attard, "Finite Strain - Isotropic Hyperelasticity," *Int. J. Solids Struct.*, vol. 40, pp. 4353–4378, 2003.

11. 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.

12. A.F. Bower, *Applied Mechanics of Solids*, CRC Press, 2009.

13. A. Gent, "A new constitutive relation for rubbers," *Rubber Chem. Technol.*, vol. 69, pp. 59–61, 1996.

14. C. Horgan and G. Saccomandi, "Constitutive Models for Compressible Nonlinearly Elastic Materials with Limiting Chain Extensibility," *J. Elasticity*, vol. 7, pp. 123–138, 2004.

15. B. Storakers, "On Material Representation and Constitutive Branching in Finite Compressible Elasticity," *J. Mech. Phys. Solids*, vol. 34, pp. 125–145, 1986.

16. Y. Gao, "Large Deformation Field Near a Crack Tip in a Rubber-like Material," *Theor. Appl. Fract. Mec.*, vol. 26, pp. 155–162, 1997.

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. Marinos, P. Marinos, 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. Zadravkovic, *Finite Element Analysis in Geothechnical Engineering*, Thomas Telford, 1999.

16. W. Tiecheng 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.

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 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 Magnetostricite 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. Lalanne 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 ff

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 behaviour of CFRP single-strap repairs”, *International Journal of Solids and Structures*, Volume 45, Issue 5, 1 March 2008, Pages 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*, Volume 37, Issue 5, May 2006, Pages 778–789.

4

Solid Mechanics

This chapter describes the Solid Mechanics interface, which is found under the **Structural Mechanics** branch () when adding a physics interface.

The Solid Mechanics Interface

The **Solid Mechanics (solid)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is intended for general structural analysis of 3D, 2D, 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 non-reflecting 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 stress** or **Plane strain** (the default). Plane stress is relevant for structures which are thin in the out-of-plane direction, such as a thin plate. Plane strain is relevant when the 2D model can be considered as a cut through an object which is long in the out-of-plane direction. For more information see the theory section.

When modeling 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 $\mathbf{u}$.

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.

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** $\mathbf{x}_{\mathrm{ref}}$ (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

To display this section, click the **Show** button ( ) and select **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 SUBMENUS

Many features for the Solid Mechanics interface are added from submenus in the Physics toolbar groups or context menu (when you right-click the node). The submenu name is the same in both cases.

The submenus at the **Domain** level are **Material Models**, **Volume Forces**, **Mass, Spring, and Damper**, 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 Pre-Tension
- Bolt Selection
- Boundary Load
- Cam-Clay Material
- Change Thickness
- Contact
- Edge Load
- 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
- Shell Connection
- Spring Foundation
- Stress Linearization
- Symmetry
- Thin Elastic Layer
- Thin-Film Damping

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:

- 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
- 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.

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

Membrane>Initial Values

Truss>Initial Values

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>Solid Mechanics>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 Change Thickness 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:

- Thermal Expansion (for materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Damping
- Viscoelasticity
- Plasticity
- 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 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

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

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/)

- 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 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
Lamé parameter μ	$\mu =$	$\frac{E}{2(1+\nu)}$	G	G	μ

TABLE 4-1: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$	$D(\lambda,\mu)$
Pressure-wave speed	c_p =	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{G(4G-E)}{\rho(3G-E)}}$	$\sqrt{\frac{K+4G/3}{\rho}}$	$\sqrt{\frac{\lambda+2\mu}{\rho}}$
Shear-wave speed	c_s =	$\sqrt{\frac{E}{2\rho(1+\nu)}}$	$\sqrt{G/\rho}$	$\sqrt{G/\rho}$	$\sqrt{\mu/\rho}$

The individual property parameters are:

- **Young's modulus** (elastic modulus) E.
- **Poisson's ratio** 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** $\mathbf{E}$, **Poisson's ratio** ν, and the **Shear modulus** $\mathbf{G}$. This defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$

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

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

Multibody Dynamics>Linear Elastic Material

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>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:

- Thermal Expansion (for materials)
- Hygroscopic Swelling
- Initial Stress and Strain
- External Stress
- External Strain
- Damping
- Viscoelasticity
- Plasticity
- 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

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.

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.

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).

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** $\varepsilon_{\mathrm{ref}}$.
  - **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** $\varepsilon_{\mathrm{ult}}$.

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[GPa])*solid.eax` $\mathrm{N/m^2}$, 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.eelvol` $\mathrm{N/m^2}$, 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

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.

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.

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.

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).

- 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

For **Neo-Hookean** you can use a mixed formulation by adding the negative mean pressure as an extra dependent variable. For this case, select the **Nearly incompressible material** check box. Then if the check box is selected, enter a value for the **Initial bulk modulus** κ.

The default values for both **Lamé parameter** λ and **Lamé parameter** μ use values **From material**.

St Venant-Kirchhoff

For **St Venant-Kirchhoff** you can use a mixed formulation by adding the negative mean pressure as an extra dependent variable. For this case, select the **Nearly incompressible material** check box.

The default values for both **Lamé parameter** λ and **Lamé parameter** μ use values **From material**.

Mooney-Rivlin, Two Parameters

For **Mooney-Rivlin, two-parameters** the **Model parameters** C_{10} and C_{01} both use values **From material**. Enter the **Initial 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**. Enter the **Initial 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**. Enter the **Initial bulk modulus** κ.

Yeoh

For **Yeoh** the **Model parameters** c_1, c_2, and c_3 all use values **From material**. Enter the **Initial bulk modulus** κ.

Ogden

For **Ogden** you can use a mixed formulation by adding the negative mean pressure as an extra dependent variable. For this case, select the **Nearly incompressible material** check box.

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** check box is selected, also enter the **Initial 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**. Enter the **Initial bulk modulus** κ.

Arruda-Boyce

For **Arruda-Boyce** you can use a mixed formulation by adding the negative mean pressure as an extra dependent variable. For this case, select the **Nearly incompressible material** check box.

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** check box is selected, also enter the **Initial 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**. Enter the **Initial 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

For **User defined** you can use a mixed formulation by adding the negative mean pressure as an extra dependent variable. In this case, select the **Nearly incompressible material** check box.

If the **Nearly incompressible material** check box is selected, enter the **Isochoric strain energy density** W_{siso} and the **Volumetric strain energy density** W_{svol}.

If the **Nearly incompressible material** check box is *not* selected, enter an expression for the **Strain energy density** W_s.

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

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** (ρ).

For entering these matrices, use the following order (Voigt notation), which is the common convention for piezoelectric materials: *xx*, *yy, zz*, *yz*, *xz*, *zy*.

The density is needed for dynamic analysis or when the elastic data is given in terms of wave speed. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use a large 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*.

The density is needed for dynamic analysis. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

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

Cam-Clay Material

The **Cam-Clay Material** node adds the equations and interface for defining the material properties for the modified Cam-clay material. This material model requires the Geomechanics Module.

The Thermal Expansion (for materials) and Initial Stress and Strain subnodes can be added to the **Cam-Clay Material**.

See also Cam-Clay Material Model in the *Structural Mechanics Theory* chapter.

MODEL INPUTS

The **Pore fluid pressure** p_{fluid} 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.

CAM-CLAY MATERIAL

To use a mixed formulation by adding the negative mean pressure as an extra dependent variable to solve for, select the **Nearly incompressible material** check box.

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, **Cam-Clay M parameter** M, **Swelling index** κ, **Compression index** λ, and **Void ratio at reference pressure** N are taken **From material**. For **User defined** enter other values or expressions.

For the **Cam-Clay M parameter** 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**.

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).

Enter a value or expression for the **Reference pressure for the parameter N** p_{refN} , the **Initial void ratio** e_0,and the **Initial consolidation pressure** pc_0.

GEOMETRIC NONLINEARITY

If a study step is geometrically nonlinear, the default behavior is to use the Green-Lagrange strain 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.

Isotropic Compression with Cam-Clay Material Model: Application Library path **Geomechanics_Module/Verification_Examples/ isotropic_compression**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Material Models>Cam-Clay Material

Ribbon

Physics tab with **Solid Mechanics** selected:

Domains>Material Models>Cam-Clay 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 and Membrane 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.

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** $C_{1\mathrm{WLF}}$. The default is 17.44.
  - **WLF constant 2** $C_{2\mathrm{WLF}}$. 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 (1)** column and the relaxation time constant τ_m in the **Relaxation time (s)** column for the spring-dashpot pair.

- Use the **Add** button (➕) to add a row to the table 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 x 10^{13} Pa·s.

In the **Shear modulus** field, enter the stiffness of the spring G_v. The default is 2 x 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 x 10^{13} Pa·s.

For large strain viscoelasticity, enter the **Relaxation time** τ_v. The default is 3000 s.

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>Hyperlastic Material>Viscoelasticity
Membrane>Linear Elastic Material>Viscoelasticity
Membrane>Nonlinear Elastic Material>Viscoelasticity
Membrane>Hyperlastic Material>Viscoelasticity

Ribbon

Physics tab with **Linear Elastic Material** or **Nonlinear Elastic Material** node selected in the model tree:

Attributes>Viscoelasticity

Plasticity

Using the **Plasticity** subnode you 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 $F(\sigma, \sigma_{\mathrm{ys}}) \leq 0$.

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** feature.
- 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 isotropic hardening** the default **Isotropic tangent modulus** $E_{T\mathrm{iso}}$ uses values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified as hardening occurs, and it is related to the effective plastic strain ε_{pe} as

$$\sigma_{ys} = \sigma_{ys0} + E_{\mathrm{iso}}\varepsilon_{pe} \text{ with } \frac{1}{E_{\mathrm{iso}}} = \frac{1}{E_{T\mathrm{iso}}} - \frac{1}{E}$$

  For the linear isotropic hardening model, the yield stress increases proportionally to the 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

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + k\varepsilon_{pe}{}^{n}$$

  the **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.

- 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

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0}\left(1 + \frac{\varepsilon_{pe}}{\varepsilon_0}\right)^{n}$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

$$\sigma_{ys}(\varepsilon_{pe}) = \sigma_{ys0} + \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{pe}}\right)$$

  the **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.

- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_∞, the **Saturation coefficient** m, and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{\mathrm{ys}}(\varepsilon_{\mathrm{pe}}) = \sigma_{\mathrm{ys0}} + (\sigma_\infty - \sigma_{\mathrm{ys0}})\left(1 - e^{-m\varepsilon^n_{\mathrm{pe}}}\right)$$

- For **User defined** isotropic hardening the **Hardening function** $\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$ uses values **From material**. The yield level σ_{ys} is modified as

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$$

  This definition implies that the hardening function $\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{pe}})$ must be zero at zero plastic strain. In other words, $\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}}$ when $\varepsilon_{\mathrm{pe}} = 0$. 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 kinematic hardening** 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:

$$\sigma_{\mathrm{b}} = C_{\mathrm{k}} \cdot \frac{2}{3}\varepsilon_{\mathrm{p}} \text{ with } \frac{1}{C_{\mathrm{k}}} = \frac{1}{E_{\mathrm{k}}} - \frac{1}{E}$$

  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

$$\dot{\sigma}_{\mathrm{b}} = \frac{2}{3}C_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{p}} - \gamma_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{pe}}\sigma_{\mathrm{b}}$$

  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:

$$\dot{\sigma}_{\mathrm{b,i}} = \frac{2}{3}C_i\dot{\varepsilon}_{\mathrm{p}} - \gamma_i\dot{\varepsilon}_{\mathrm{pe}}\sigma_{\mathrm{b,i}}$$

The total back stress σ_{b} is then computed from the sum

$$\sigma_{\mathrm{b}} = \frac{2}{3}C_0\varepsilon_{\mathrm{p}} + \sum_{\mathrm{i}=1}^{N}\sigma_{\mathrm{b,i}}$$

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 **Hyperlastic 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 and Membrane 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.

CREEP DATA

Nonlinear Structural Materials Module

Select a **Material model** — **Norton**, **Norton-Bailey**, **Garofalo (hyperbolic sine)**, **Navarro-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** $\varepsilon_{\mathrm{shift}}$, 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.

Navarro-Herring

For **Navarro-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 **Creep rate** F_{cr}.

Volumetric

For **Volumetric** enter the **Creep rate** F_{cr}.

User defined

For **User defined** enter each element for the symmetric **Creep rate tensor** F_{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

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.

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** a.
- **Deformation resistance sensitivity** n.

Each of the material properties can either be defined obtained **From material,** or as **User defined**. In the latter case, enter a value or an expression.

Viscoplastic Creep in Solder Joints: Application Library path **Nonlinear_Structural_Materials_Module/Viscoplasticity/ viscoplastic_solder_joints**

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 $F(\sigma, \sigma_{\mathrm{ys}}) \leq 0$.

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 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** σ_{ys0} uses values **From material** and represents the stress level where viscoplastic deformation starts.

Isotropic Hardening Model

Select the type of linear or nonlinear isotropic hardening model from the **Isotropic hardening model** list.

- Select **Perfectly plastic** (ideal viscoplasticity) if the material can undergo viscoplastic deformation without any increase in yield stress.
- For **Linear isotropic hardening** the default **Isotropic tangent modulus** $E_{T\mathrm{iso}}$ uses values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified as hardening occurs, and it is related to the effective viscoplastic strain $\varepsilon_{\mathrm{vpe}}$ as

  $$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + E_{\mathrm{iso}}\varepsilon_{\mathrm{vpe}} \text{ with } \frac{1}{E_{\mathrm{iso}}} = \frac{1}{E_{T\mathrm{iso}}} - \frac{1}{E}$$

  For the linear isotropic hardening model, the yield stress increases proportionally to the effective viscoplastic strain $\varepsilon_{\mathrm{vpe}}$. The Young's modulus E is taken from the elastic material properties.
- Select **Ludwik** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the power-law

  $$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + k\varepsilon_{\mathrm{vpe}}{}^{n}$$

  the **Strength coefficient** k and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**.
- For **Swift** nonlinear isotropic hardening, the **Reference strain** ε_0 and the **Hardening exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is modified by the power-law

  $$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}}\left(1 + \frac{\varepsilon_{\mathrm{vpe}}}{\varepsilon_0}\right)^{n}$$

- Select **Voce** from the list to model nonlinear isotropic hardening. The yield level σ_{ys} is modified by the exponential law

  $$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + \sigma_{sat}\left(1 - e^{-\beta\varepsilon_{\mathrm{vpe}}}\right)$$

  the **Saturation flow stress** σ_{sat} and the **Saturation exponent** β use values **From material** (if it exists) or **User defined**.
- For **Hockett-Sherby** nonlinear isotropic hardening, the **Steady-state flow stress** σ_{∞}, the **Saturation coefficient** m, and the **Saturation exponent** n use values **From material** (if it exists) or **User defined**. The yield level σ_{ys} is increased by the exponential law

$$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + (\sigma_\infty - \sigma_{\mathrm{ys0}})\left(1 - e^{-m\varepsilon^n_{\mathrm{vpe}}}\right)$$

- For **User defined** isotropic hardening the **Hardening function** $\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{vpe}})$ uses values **From material**. The yield level σ_{ys} is modified as

  $$\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}} + \sigma_{\mathrm{h}}(\varepsilon_{\mathrm{vpe}})$$

  This definition implies that the hardening function $\sigma_{\mathrm{h}}(\varepsilon_{\mathrm{vpe}})$ must be zero at zero viscoplastic strain. In other words, $\sigma_{\mathrm{ys}} = \sigma_{\mathrm{ys0}}$ when $\varepsilon_{\mathrm{vpe}} = 0$. 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 kinematic hardening** 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:

  $$\sigma_{\mathrm{b}} = C_{\mathrm{k}} \cdot \frac{2}{3}\varepsilon_{\mathrm{vp}} \quad \text{with} \quad \frac{1}{C_{\mathrm{k}}} = \frac{1}{E_{\mathrm{k}}} - \frac{1}{E}$$

  This is Prager's linear kinematic hardening model, so the back stress σ_{b} is collinear to the viscoplastic strain tensor $\varepsilon_{\mathrm{vp}}$.

- 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

  $$\dot{\sigma}_{\mathrm{b}} = \frac{2}{3}C_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{vp}} - \gamma_{\mathrm{k}}\dot{\varepsilon}_{\mathrm{vpe}}\sigma_{\mathrm{b}}$$

  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:

$$\dot{\sigma}_{\mathrm{b,i}} = \frac{2}{3}C_i\dot{\varepsilon}_{\mathrm{vp}} - \gamma_i\dot{\varepsilon}_{\mathrm{vpe}}\sigma_{\mathrm{b,i}}$$

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

$$\sigma_{\mathrm{b}} = \frac{2}{3}C_0\varepsilon_{\mathrm{vp}} + \sum_{\mathrm{i}=1}^{N}\sigma_{\mathrm{b,i}}$$

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

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 obtained **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 the choose to use there 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** α 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** $\varepsilon_{\mathrm{pvol,max}}$, 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** $\varepsilon_{\mathrm{pvol,max}}$, 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/).

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.

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

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

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

$$\varepsilon_{\mathrm{th}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\mathrm{t}}(\tau)d\tau\right) - 1$$

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 non-isotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

Enter a value or expression for the **Strain reference temperature** T_{ref} which is the reference temperature that defines the change in temperature together with the actual temperature.

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**.

- Thermoelastic Damping
- Entropy and Thermoelasticity

Thermal Stresses in a Layered Plate: Application Library path **Structural_Mechanics_Module/Thermal-Structure_Interaction/layered_plate** **MEMS_Module/Actuators/layered_plate**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Linear Elastic Material>Thermal Expansion
Solid Mechanics>Nonlinear Elastic Material>Thermal Expansion
Solid Mechanics>Hyperelastic Material>Thermal Expansion
Solid Mechanics>Piezoelectric Material>Thermal Expansion
Solid Mechanics>Cam-Clay 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

$$\varepsilon_{\mathrm{hs}} = \beta_{\mathrm{h}}(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration.

The **Hygroscopic Swelling** subnode is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

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 `1.5e-4` m^3/kg. When a non-isotropic 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

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.

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

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.

EXTERNAL STRESS

Select a **Stress input** — **Stress tensor (Material)**, **Stress tensor (Spatial)** or **Pore pressure**.

- 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** $\mathrm{S}_{\mathrm{ext}}$ as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type** — **Add to stress tensor** 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** α_{B} to specify the fraction of the pore pressure to use. As a default, its value is taken **From material**. Choose **User defined** to enter another value or expression. When using **Pore pressure**, there is no contribution to the stress tensor, the only effect of the pressure is as a load.

- 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
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.

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 the **Hyperelastic Material**.

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-Kirchoff 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 $\varepsilon_{\mathrm{ext}}$. 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 $\boldsymbol{F}_{\mathrm{ext}}$. From the **External deformation gradient** list, you can choose **User defined**, or any deformation gradient tensor which is announced by another physics interface. If you select **User defined**, enter values or expressions for the components of the deformation gradient tensor.

Deformation gradient, inverse

For **Deformation gradient, inverse**, enter an inelastic inverse deformation gradient contribution $\boldsymbol{F}_{\mathrm{ext}}^{-1}$. 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

$$\varepsilon_{\mathrm{ext}} = \frac{1}{2}(F^T_{\mathrm{ext}}F_{\mathrm{ext}} - I)$$

- If you select **Deformation gradient** form the **Strain input** list when engineering strains are expected, the engineering strain tensor is computed from

$$\varepsilon_{\mathrm{ext}} = \frac{1}{2}(F_{\mathrm{ext}} + F^T_{\mathrm{ext}}) - I$$

  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

$$F_{\mathrm{ext}} = I + \varepsilon_{\mathrm{ext}}$$

See also 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

Ribbon

Physics tab with **Linear Elastic Material**, **Nonlinear Elastic Material**, or **Hyperelastic Material** node selected in the model builder tree:

Attributes>External Strain

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 Nonlinear Elastic Material. Four different variables describing the failure risk will be 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 will 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/)

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	MEMBRANE	BEAM	TRUSS
von Mises	X	X	X	X	X
Tresca	X	X	X	X	X
Rankine	X	X	X	X	X
St. Venant	X	X	X	X	X
Mohr-Coulomb	X	-	-	-	-
Drucker-Prager	X	-	-	-	-
Jenkins	X	X	X	-	-
Waddoups	X	X	X	-	-
Azzi-Tsai-Hill	Plane stress	X	-	-	-
Norris	Plane stress	X	-	-	-
Modified Tsai-Hill	Plane stress	X	-	-	-
Tsai-Hill	X	X	X	-	-
Hoffman	X	X	-	-	-
Tsai-Wu Orthotropic	X	X	-	-	-
Tsai-Wu Anisotropic	X	X	-	-	-
User defined	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 **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

Membrane>Linear Elastic Material>Safety

Membrane>Nonlinear Elastic Material>Safety

Shell>Linear Elastic Material>Safety

Plate>Linear 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

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.

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/).

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
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

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

$$\xi = \alpha_{\mathrm{dM}} m + \beta_{\mathrm{dK}} k$$

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** $\alpha_{\mathrm{d}M}$ and the **Stiffness damping parameter** $\beta_{\mathrm{d}K}$.

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

$$\alpha_{\mathrm{dM}} = 4\pi f_1 f_2 \frac{\varsigma_1 f_2 - \varsigma_2 f_1}{f_2^2 - f_1^2}$$

$$\beta_{\mathrm{dK}} = \frac{\varsigma_2 f_2 - \varsigma_1 f_1}{\pi(f_2^2 - f_1^2)}$$

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_{DVo}. The orientations are the same as in the parent node.

When **Anisotropic loss factor** is selected, use the **Loss factor for elasticity matrix D** list to select the way to enter η_D or η_{DVo}. The default is to take the values **From material**. For **User defined** enter the components of η_D or η_{DVo} in the upper-triangular part of a symmetric 6-by-6 matrix.

The values for the loss factors are ordered in two different ways, consistent with the selection of either **Standard (XX, YY, ZZ, XY, YZ, XZ)** or **Voigt (XX, YY, ZZ, YZ, XZ, XY)** notation in the corresponding Linear Elastic Model. If the values are taken from the material, these loss factors are found in the **Anisotropic** or **Anisotropic, Voigt notation** property group for the material. For an isotropic material, the anisotropic loss factor is always given as η_D using the standard notation.

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_{GVo}. 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

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, either via using the loss factor material data for the stiffness, or in form of the 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

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.

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** $\alpha_{\mathrm{d}M}$ and the **Stiffness damping parameter** $\beta_{\mathrm{d}K}$.

In this damping model, the damping parameter ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{\mathrm{dM}}m + \beta_{\mathrm{dK}}k$$

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

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree: **Attributes>Mechanical Damping**

Coupling Loss

The **Coupling Loss** subnode allows you to model losses in the piezoelectric coupling in a Piezoelectric Material, either by using the loss factor material data for the coupling matrix or as Rayleigh proportional damping.

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** $\Delta\varepsilon_{rS}$ 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_T, 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

Ribbon

Physics tab with **Piezoelectric Material** node selected in the model tree:
Attributes>Conduction Loss

External Stress-Strain Relation

The **External Stress-Strain Relation** is a special type of material model where the computation of second Piola-Kirchoff 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 (2D and 3D).
- 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/).

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.

The density is needed for dynamic analysis. It is also used when computing mass forces for gravitational or rotating frame loads, and when computing mass properties (Computing Mass Properties).

If the density is set to zero, the center of mass of the rigid domain will be the same as if a uniform non-zero density had been given.

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 X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{P}} = \mathbf{X}_{\mathrm{P,input}} + \mathbf{X}_{\mathrm{offset}}$$

Rigid Domain Material Model

Modeling Rigid Bodies: Application Library path **Structural_Mechanics_Module/Connectors_and_Mechanisms/rigid_domain**

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Material Models>Rigid Domain

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** 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 X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{P}} = \mathbf{X}_{\mathrm{P,input}} + \mathbf{X}_{\mathrm{offset}}$$

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.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Rigid Domain>Fixed Constraint

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 the 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 X_{offset}.

The center of rotation used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{P}} = \mathbf{X}_{\mathrm{P,input}} + \mathbf{X}_{\mathrm{offset}}$$

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

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 t 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)**, $\mathbf{X}_\mathrm{p}$, 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 X_offset.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_\mathrm{P} = \mathbf{X}_\mathrm{P,input} + \mathbf{X}_\mathrm{offset}$$

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

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 $\mathbf{M}$.
- 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
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**, $\mathbf{X}_\mathrm{m}$, 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 X_offset.

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{m}} = \mathbf{X}_{\mathrm{m,input}} + \mathbf{X}_{\mathrm{offset}}$$

MASS AND MOMENT OF INERTIA

Enter values or expressions for the **Mass** m. Then for the **Moment of inertia** the axis directions of the moment of inertia tensor are given by the selected coordinate system.

- For 3D components, select **Isotropic** (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

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 X_{offset}.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{P}} = \mathbf{X}_{\mathrm{P,input}} + \mathbf{X}_{\mathrm{offset}}$$

SPRING

Select a **Spring type** — **Spring constant** or **Force as function of extension**.

When **Spring constant** is selected, the translational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **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 $\mathbf{F}_\mathrm{s}$. It must be a 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 $\mathbf{M}_\mathrm{s}$. It must be a 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** $\eta_{\mathrm{u,s}}$, 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** $\eta_{\mathrm{u,k}}$ 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 selecting **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 selecting **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

Multibody Dynamics>Rigid Domain>Spring Foundation

Shell>Rigid Domain>Spring Foundation

Beam>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 and Multibody Dynamics interfaces, the **Free** node applies to boundaries.
- For the 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

Membrane>Free

Beam>Free

Truss>Free

Multibody Dynamics>Free

Ribbon

Physics tab with a physics interface selected:

Boundaries>Solid Mechanics>Free

Boundaries>Multibody Dynamics>Free

Edges>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.

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

$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which force the domain to move only diagonally in the xy-plane.

CONSTRAINT SETTINGS

To display this section, click the **Show** 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

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.

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)

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 **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.

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)

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 **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. If there are rotational degrees of freedom, they will also be zero.

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..

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

Shell>Face Constraints>Fixed Constraint

Shell>Fixed Constraint (Edge)

Shell>Points>Fixed Constraint

Plate>Face Constraints>Fixed Constraint

Plate>Fixed Constraint (Boundary)

Plate>Points>Fixed Constraint

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 **Shell** selected:

Boundaries>Face Constraints>Fixed Constraint

Edges>Shell>Fixed Constraint

Points>Shell>Fixed Constraint

Physics tab with **Plate** selected:

Domains>Face Constraints>Fixed Constraint

Boundaries>Plate>Fixed Constraint

Points>Plate>Fixed Constraint

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

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

The **Thermal Expansion** subnode is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

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.
- The virtual surrounding material 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 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 non-isotropic 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 **Strain 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

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 equivalent to a Symmetry condition.

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

Solid Mechanics>Roller

Multibody Dynamics>Roller

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries>Solid Mechanics>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.

The **Symmetry** node is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

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

Membrane>More Constraints>Symmetry

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries>More Constraints>Symmetry

Physics tab with **Membrane** selected:

Edges>More Constraints>Symmetry

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.

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

Membrane>More Constraints>Antisymmetry

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries>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.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

- Rigid body motion in the Structural Mechanics Modeling chapter.
- Rigid Motion Suppression Boundary Condition in the Structural Mechanics Theory chapter.

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>Rigid Body Suppression

Shell>Face Constraints>Rigid Body Suppression

Plate>Face Constraints>Rigid Body Suppression

Membrane>Face Constraints>Rigid Body Suppression

Beam>Line Constraints>Rigid Body Suppression

Multibody Dynamics>Domain Constraints>Rigid Body Suppression

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Domains>Domain Constraints>Rigid Body Suppression

Physics tab with **Shell** or **Membrane** selected:

Boundaries>Face Constraints>Rigid Body Suppression

Physics tab with **Plate** selected:

Domains>Face Constraints>Rigid Body Suppression

Physics tab with **Beam** or **Truss** selected:

Edges>Line Constraints>Fixed Constraint

Body Load

Add a **Body Load** to domains for modeling volumetric loads.

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	$\mathbf{F}_V$	N/m^3	domains	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per unit area	$\mathbf{F}_A$	N/m^2	domains	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	domains	3D (x, y, z) 2D (x, y) 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

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

Physics tab with **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 $\mathbf{g} = \rho g \mathbf{e}_\mathrm{g}$ 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/).

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 $9.8066\ \mathrm{m/s^2}$.

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

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** 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/).

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** $\mathbf{r}_{\mathrm{bp}}$ and **Rotation axis direction** $\mathbf{e}_{\mathrm{ax}}$.

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**, **RPM**, or **Revolutions per time**. Then enter a value as needed for **Angular velocity magnitude** Ω, **RPM**, 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

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** 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.

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 load current surface normal is 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	$\mathbf{F}_A$	N/m^2	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Force per unit length	$\mathbf{F}_L$	N/m	boundaries	2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	boundaries	3D (x, y, z) 2D (x, y) 2D axisymmetric (r, z)
Pressure	p	Pa	boundaries	3D (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

Multibody Dynamics>Boundary Load

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Boundaries>Solid Mechanics>Body Load

Boundaries>Multibody Dynamics>Body Load

Edge Load

Add an **Edge Load** to 3D components to apply a force distributed along an edge.

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 $\mathbf{F}_\mathrm{L}$.
- The total force $\mathbf{F}_\mathrm{tot}$. 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

Multibody Dynamics>Edges>Edge Load

Ribbon

Physics tab with **Solid Mechanics** or **Multibody Dynamics** selected:

Edges>Edge Load

Point Load

Add a **Point Load** to points for concentrated forces at points in 2D and 3D.

FORCE

Enter values or expressions for the components of the point load $\mathbf{F}_\mathrm{P}$.

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

Multibody Dynamics>Points>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 **Truss** selected:

Points>Truss>Point Load

Ring Load

Add a **Ring Load** to points located at $R > 0$ in axially symmetric models. Select this feature from the **Points** submenu.

FORCE

Select the **Load type** — **Force per unit length** (the default) or **Total force.** Enter values or expressions for $\mathbf{F}_{\mathrm{L}}$ or $\mathbf{F}_{\mathrm{tot}}$.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.
- 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 non-zero radial coordinates actually represent a Ring Load.

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)

Membrane>Point Load (on Axis)

Ribbon

Physics tab with **Solid Mechanics** selected:

Points>Point Load (on Axis)

Physics tab with **Membrane** selected:

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 non-zero 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/).

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	$\mathbf{k}_V$	$N/(m·m^3)$	domains	3D, 2D, and 2D axisymmetric
Total spring constant	$\mathbf{k}_{tot}$	N/m	domains, edges	3D, 2D, and 2D axisymmetric
Spring constant per unit area	$\mathbf{k}_A$	$N/(m·m)^2$	boundaries	3D, 2D
Spring constant per unit length	$\mathbf{k}_L$	N/(m·m)	edges, boundaries (2D)	3D, 2D
Spring constant	$\mathbf{k}_P$	N/m	points	3D, 2D, and 2D axisymmetric
Force per volume as function of extension	$\mathbf{F}_V$	N/m^3	domains	3D, 2D, and 2D axisymmetric
Total force as function of extension	$\mathbf{F}_{tot}$	N	domains, boundaries, edges	3D, 2D, and 2D axisymmetric
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	domains, boundaries	3D, 2D
Force per length as function of extension	$\mathbf{F}_L$	N/m	edges	3D
Force as function of extension	$\mathbf{F}_P$	N	points	3D, 2D, and 2D axisymmetric
Use material data	N/A	N/A	boundaries	3D, 2D, and 2D axisymmetri

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	$\frac{E}{2G}-1$	$\frac{1}{2}\left(1-\frac{3G}{3K+G}\right)$

TABLE 4-6: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(E,G)$	$D(K,G)$
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 selecting **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	$\mathbf{d}_\mathrm{V}$	N·s/(m·m^3)	domains, boundaries (2D)	3D, 2D
Damping constant per unit area	$\mathbf{d}_\mathrm{A}$	N·s/(m·m^2)	domains, boundaries	3D, 2D, and 2D axisymmetric
Total damping constant	$\mathbf{d}_\mathrm{tot}$	N·s/m	domains, boundaries, edges, points	3D, 2D, and 2D axisymmetric
Damping constant per unit length	$\mathbf{d}_\mathrm{L}$	N·s/(m·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

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 **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 non-zero 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/).

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	$\mathbf{k}_{\mathrm{tot}}$	N/m	3D, 2D, and 2D axisymmetric
Spring constant per unit area	$\mathbf{k}_A$	N/(m·m)^2	3D, 2D, and 2D axisymmetric
Spring constant per unit length	$\mathbf{k}_L$	N/(m·m)	2D
Total force as function of extension	$\mathbf{F}_{\mathrm{tot}}$	N	3D, 2D, and 2D axisymmetric
Force per area as function of extension	$\mathbf{F}_A$	N/m^2	3D, 2D, and 2D axisymmetric
Force per length as function of extension	$\mathbf{F}_L$	N/m	2D
Use material data	N/A	N/A	3D, 2D, and 2D axisymmetric

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 selecting **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	$\mathbf{d}_A$	N·s/(m·m^2)	3D, 2D, and 2D axisymmetric
Total damping constant	$\mathbf{d}_{tot}$	N·s/m	3D, 2D, and 2D axisymmetric
Damping constant per unit length	$\mathbf{d}_L$	N·s/(m·m)	2D

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

Ribbon

Physics tab with **Solid Mechanics** selected:

Boundaries>Mass, Spring, and Damper>Thin Elastic Layer

Pairs>Solid Mechanics>Thin Elastic Layer

Predeformation

Use the **Predeformation** subnode to specify that the elastic forces in Spring Foundation or Thin Elastic Layer are non-zero at zero displacement. Thus you can model cases where the unstressed state of the spring is in another configuration than the one described by the geometry.

The value of the predeformation can vary during the simulation, so it should not be interpreted as an initial value.

SPRING PREDEFORMATION

Based on space dimension, enter the values for the **Spring Predeformation** $\mathbf{u}_0$.

> You can assign the load caused by the predeformation to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Spring Foundation>Predeformation

Solid Mechanics>Thin Elastic Layer>Predeformation

Shell>Spring Foundation>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** or **Thin Elastic Layer** node selected in the model tree: **Attributes>Predeformation**

Added Mass

The **Added Mass** node is available on domains, 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 **Added Mass** node is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

COORDINATE SYSTEM SELECTION

The added mass values are given with respect to the selected coordinate directions.

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
Mass per unit area	ρ_A	kg/m^2	domains, boundaries
Mass per unit length	ρ_L	kg/m	edges
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.

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)

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 **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.

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; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = \mathbf{u}(\mathbf{x}_\mathrm{s})$. If the source and destination boundaries are rotated with respect to each other, a transformation is automatically performed, so that corresponding displacement components are connected.
- For **Antiperiodicity** the displacements on the destination are set equal to the displacements on the source with the sign reversed; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = -\mathbf{u}(\mathbf{x}_\mathrm{s})$. If the source and destination boundaries are rotated with respect to each other, a transformation is automatically performed, so that corresponding displacement components are connected.
- For **Floquet periodicity** enter a **k-vector for Floquet periodicity** $\mathbf{k}_\mathrm{F}$. This is the wave number vector for the excitation.
- For **Cyclic symmetry** 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 $u_i(\mathbf{x}_\mathrm{d}) = u_i(\mathbf{x}_\mathrm{s})$ or $u_i(\mathbf{x}_\mathrm{d}) = -u_i(\mathbf{x}_\mathrm{s})$ 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*.

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

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** $\mathbf{d}_\mathrm{i}$. The defaults for all values are `0.5*solid.rho*(solid.cp+solid.cs)`.

Low-Reflecting Boundary Condition

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>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. The selected boundaries 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 Beam 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.

The **Rigid Connector** node is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

! In versions prior to 4.4, it was also possible to add a **Rigid Domain** subnode to the **Rigid Connector**. This node still appears and functions when you open an old model where it was used, but it is not possible add a new one.

The functionality has been superseded by the more powerful concept of a Rigid Domain as a material model.

For information about the old functionality, see the documentation for version 4.3b or earlier.

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 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.

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.

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 geometrical objects.
- 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 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.

$$\mathbf{X}_{\mathrm{C}} = \mathbf{X}_{\mathrm{C,input}} + \mathbf{X}_{\mathrm{offset}}$$

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** $u_{0\mathrm{x}}$
- **Prescribed in y direction** $u_{0\mathrm{y}}$
- *For 3D components:* **Prescribed in z direction** $u_{0\mathrm{z}}$

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

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
Multibody Dynamics>Rigid Connectior

Ribbon

Physics tab with **Solid Mechanics** selected:
Boundaries>Connections>Rigid Connector

Physics tab with **Multibody Dynamics** selected:
Boundaries>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

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

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

non-isotropic 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 **Strain 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

Multibody Dynamics>Rigid connector>Thermal Expansion

Ribbon

Physics tab with **Rigid connector** 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)**, $\mathbf{X}_{\mathrm{p}}$, 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 X_{offset}.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_{\mathrm{P}} = \mathbf{X}_{\mathrm{P,input}} + \mathbf{X}_{\mathrm{offset}}$$

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

This section is only available for 3D components. The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. Select a **Coordinate system** for specifying the directions of the moment.

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**, $\mathbf{X}_\mathrm{m}$, 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 X_offset.

The center of mass used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_\mathrm{m} = \mathbf{X}_\mathrm{m,input} + \mathbf{X}_\mathrm{offset}$$

MASS AND MOMENT OF INERTIA

Enter values or expressions for the **Mass** m. Also enter values or expressions for the **Moment of inertia** $\mathbf{I}$. The axis directions of the moment of inertia tensor are given by the coordinate system selection in the parent Rigid Connector node.

- For 3D components, select **Isotropic** (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)**, $\mathbf{X}_\mathrm{p}$, 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 X_offset.

The location used is the sum of the vector obtained from any of the input methods and the offset vector.

$$\mathbf{X}_\mathrm{P} = \mathbf{X}_\mathrm{P,input} + \mathbf{X}_\mathrm{offset}$$

SPRING

Select a **Spring type** — **Spring constant** or **Force as function of extension**.

When **Spring constant** is selected, the translational spring matrix can be entered as **Isotropic**, **Diagonal**, **Symmetric**, or **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 $\mathbf{F}_\mathrm{s}$. It must be a 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 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 selecting **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 selecting **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. All the selected boundaries 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 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 Attachments is included in the *Multibody Dynamics Module User's Guide*.

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

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

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain in the attachment.

Select **Inherit from domain** to take the thermal expansion data from the domain to which it is attached. This should only be used when:

- The temperature and the thermal expansion coefficient do not have a spatial variation.
- The virtual material in the attachment has the same thermal expansion as the domain itself.

When **Inherit from domain** is not selected, enter:

- A value or expression for **Temperature** T, specifying the temperature distribution of the attachment. Any spatial variation must be an explicit function of the material frame coordinates. It is not possible to use a computed temperature distribution.
- The **Coefficient of thermal expansion** 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

non-isotropic 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 **Strain 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

Shell 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.Eequ`, 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.Eequ` 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 towards 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 en 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)*solid.<contact_tag>.E_char/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** $d_{\mathrm{offset,d}}$. 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** $d_{\mathrm{offset,s}}$. 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**. To display this section, click the **Show** button ( ) and select **Discretization**. 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** $\Delta_{\mathrm{contact}}$.

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

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.

FRICTION

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** $T_{t,\mathrm{max}}$. 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_{dcf}.

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^niterCMP,1)*solid.<contact_tag>.E_char/3/solid.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 `solid.<contact_tag>.E_char/3/solid.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** $\mathbf{x}_{\mathrm{m,\,old}}$. These serve as initial values to compute the tangential slip. The default value is (X, Y, Z) and indicates that the contacting boundaries are perfectly coincident in the initial state. The mapped source coordinates are defined as the location on the source boundary where it is hit by a certain point on the destination boundary.

The entries in this section should be given as components in the material frame.

DISCRETIZATION

This section can only be displayed if **Tangential Force Method** is **Augmented Lagrangian**. To display it, click the **Show** button (👁) and select **Discretization**. 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** $\Delta_{\mathrm{contact}}$, which is the gap when the friction becomes active.

You can also request that the total accumulated slip distance it 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:
Atributes>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 $p_{\mathrm{n}0}$, at which adhesion is initiated.

When the criterion is **Gap**, enter the distance between the source and destination boundaries δ_0, at which adhesion is initiated. A negative value indicates that there must be a certain 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, v, 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_{IIc}. 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.

- 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:

Atributes>Adhesion

Bolt Pre-Tension

Use the **Bolt Pre-Tension** node to define the pre-stress force in pre-tensioned bolts. It is available for 3D models 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 Pre-Tension** node is only available with some COMSOL products (see http://www.comsol.com/products/specifications/).

BOLT PRE-TENSION

Select a **Pre-tension type** — **Pre-tension force** (the default) or **Pre-tension stress**.

- For **Pre-tension force** enter a value or expression for F_p, the pre-tension force in the bolt.
- For **Pre-tension stress** enter a value or expression for σ_p, the pre-tension stress in the bolt. The pre-tension force is computed through multiplication by the actual area of each selected bolt.

- Pre-tensioned Bolts in the Structural Mechanics Modeling chapter for an complete description of how to model pre-tensioned bolts.
- Bolt Pre-Tension Study for information about the **Bolt Pre-Tension** study type.

SOLVER SUGGESTION

This section is present only in models created in versions prior to 5.3, in which the new **Bolt Pre-Tension** study type was introduced. The purpose is to maintain backwards compatibility for older models.

Select **Solve in bolt pre-tension study only** check box to make the bolt pre-tension degrees of freedom behave as in a version 5.3 model, that is being solved for only in a study step of the **Bolt Pre-Tension** type. By default, the check box is cleared, and then the bolt pre-tension degrees of freedom are solved for in any type of study step, unless you explicitly suppress that.

The settings of the **Solve in bolt pre-tension study only** check box only affects new study sequences being generated. Existing study sequences will keep the current state for the bolt pre-tension degrees of freedom.

Once you select the **Solve in bolt pre-tension 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 Pre-Tension** node from now on assumes the version 5.3 behavior.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Bolt Pre-Tension

Ribbon

Physics tab with **Solid Mechanics** selected in the Model Builder tree:

Global>Bolt Pre-Tension

Bolt Selection

The **Bolt Selection** subnode is automatically added as a default node to the Bolt Pre-Tension 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.

BOLT SELECTION

Enter a **Bolt label**. The default is `Bolt_1`. The label is used for identification during postprocessing.

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 pre-tension load in the parent node is by **Pre-tension 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.

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.

Pre-tensioned Bolts in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Solid Mechanics>Bolt Pre-Tension>Bolt Selection

Ribbon

Physics tab with **Bolt Pre-Tension** selected in the Model Builder tree:
Attributes>Bolt 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:

$$F_{\mathrm{freq}} = F(f) \cdot \cos(2\pi ft + \phi)$$

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>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

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 non-zero 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 Velocity, Prescribed Acceleration, 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.

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

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.

- 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.el11`	Strain tensor, local coordinate system, 11 component	All materials[1]	Total strain in the local directions.
`solid.eel11`	Elastic strain tensor, local coordinate system, 11 component	All materials[1]	

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.Sl11`	Second Piola-Kirchhoff stress, local coordinate system, 11 component	All materials[1]	
`solid.SdevX`	Deviatoric second Piola-Kirchhoff stress, X component	All material[1]	See also Invariants of the Stress Tensor.
`solid.PxX`	First Piola-Kirchhoff stress, xX component	All materials[1]	See also Defining Stress.

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**.

5

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 theory for the Shell and Plate interfaces is discussed in this section:

- About Shells and Plates
- Theory Background for the Shell and Plate Interfaces
- Reference for the Shell Interface

About Shells and Plates

A *shell* is a thin-walled structure in 3D 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 or on the boundary of a solid 3D object. In the latter 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. It has an MITC formulation where MITC means *mixed interpolation of* *tensorial components*. A general description of this element family is 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.

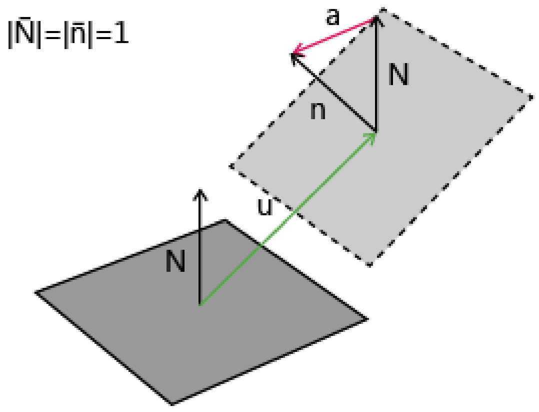


Figure 5-1: The degrees of freedom in the shell interface. **N** *is the normal vector in the original configuration and* **n** *is the normal in the deformed state.*

The degrees of freedom represent the displacements on the *reference surface*. 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, **u**, can be expressed as

$$\mathbf{u} = \mathbf{u}_\mathrm{R} + \zeta_0\mathbf{a}$$

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

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 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 Shell interface changes to (X, Y, Z).

Theory Background for the Shell and Plate Interfaces

Several topics are discussed in this section:

- Geometry and Deformation
- Strains
- Offset
- Rotation Representation
- The MITC Shell Formulation
- Initial Values and Prescribed Values
- Symmetry and Antisymmetry Boundary Conditions
- External Loads
- Stress and Strain Calculations
- Local Coordinate Systems
- Connection Between Shells and Solids
- Connection Between Shells and Beams

GEOMETRY AND DEFORMATION

Let **r** be the undeformed shell midsurface position, ξ^i be element local (possibly nonorthogonal) coordinates with origin in the shell midsurface, and **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

$$|\mathbf{n}+\mathbf{a}| = 1 \qquad (5\text{-}1)$$

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 **r**, **u**, **n**, and **a** are interpolated by the nth-order Lagrange basis functions. The basic assumption is that the position of a point within the shell after deformation has a linear dependence of the thickness coordinate, and thus is

$$\mathbf{x}(\xi^1,\xi^2,\xi^3) = \mathbf{r}(\xi^1,\xi^2)+\mathbf{u}(\xi^1,\xi^2)+\xi^3(\mathbf{n}(\xi^1,\xi^2)+\mathbf{a}(\xi^1,\xi^2))$$

The superscripts indicate contravariant indices, while subscripts indicate covariant indices.

STRAINS

The in-plane Green-Lagrange strain in the local covariant components can then be written as

$$\varepsilon_{\alpha\beta} = \frac{1}{2}\Bigg[\frac{\partial}{\partial\xi^\alpha}(\mathbf{r}+\mathbf{u}+\xi^3(\mathbf{n}+\mathbf{a}))\cdot\frac{\partial}{\partial\xi^\beta}(\mathbf{r}+\mathbf{u}+\xi^3(\mathbf{n}+\mathbf{a}))-\frac{\partial}{\partial\xi^\alpha}(\mathbf{r}+\xi^3\mathbf{n})\cdot\frac{\partial}{\partial\xi^\beta}(\mathbf{r}+\xi^3\mathbf{n})\Bigg] = \gamma_{\alpha\beta}+\xi^3\chi_{\alpha\beta}+(\xi^3)^2\kappa_{\alpha\beta}$$

The indices α and β range from 1 to 2. The transverse shear strains in local covariant components are

$$\varepsilon_{3\alpha} = \varepsilon_{\alpha 3} =$$

$$\frac{1}{2}\left[\frac{\partial}{\partial\xi^{\alpha}}(\mathbf{r}+\mathbf{u}+\xi^{3}(\mathbf{n}+\mathbf{a}))\cdot(\mathbf{n}+\mathbf{a})-\frac{\partial}{\partial\xi^{\alpha}}(\mathbf{r}+\xi^{3}\mathbf{n})\cdot\mathbf{n}\right] = \zeta_{\alpha}+\xi^{3}\omega_{\alpha}$$

The constitutive relation for the shell elements is a plane stress assumption, as is customary in shell theory. The strain component in the normal direction ε_{33} is thus irrelevant. The different parts of the strain tensors above can be written out as

$$\gamma_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{r}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{r}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{u}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{u}}{\partial\xi^{\beta}}\right]$$

$$\chi_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial\mathbf{r}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{a}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{r}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{n}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{n}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{u}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{a}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{u}}{\partial\xi^{\beta}}\right]$$

$$\kappa_{\alpha\beta} = \frac{1}{2}\left[\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{n}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{n}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{a}}{\partial\xi^{\beta}}+\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\frac{\partial\mathbf{a}}{\partial\xi^{\beta}}\right]$$

$$\zeta_{\alpha} = \frac{1}{2}\left[\frac{\partial\mathbf{r}}{\partial\xi^{\alpha}}\cdot\mathbf{a}+\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\cdot\mathbf{n}+\frac{\partial\mathbf{u}}{\partial\xi^{\alpha}}\cdot\mathbf{a}\right]$$

$$\omega_{\alpha} = \frac{1}{2}\left[\frac{\partial\mathbf{n}}{\partial\xi^{\alpha}}\cdot\mathbf{a}+\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\cdot\mathbf{n}+\frac{\partial\mathbf{a}}{\partial\xi^{\alpha}}\cdot\mathbf{a}\right]$$

In a geometrically linear analysis, the nonlinear terms (products between $\mathbf{u}$, $\mathbf{a}$, and their derivatives) disappear. In all study types, the contributions from the parts $\kappa_{\alpha\beta}$ and ω_{α} are ignored. They are small unless the element has an extremely high ratio between thickness and radius of curvature, in which case the errors from using shell theory are large anyway.

Note that ζ_{α} here is a strain component, not to be confused with the local coordinate in the normal direction or offset.

OFFSET

It is possible to model a shell with a midsurface that is not located at the meshed surface but at a certain offset from it. The offset is assumed to occur along the normal of the shell surface. In this case,

$$\mathbf{r}(\xi^{1},\xi^{2}) = \mathbf{r}_{\mathrm{R}}(\xi^{1},\xi^{2})+\zeta_{o}\mathbf{n}(\xi^{1},\xi^{2})$$

where $\mathbf{r}_\mathrm{R}$ is the position of the meshed *reference surface* and ζ_o is the offset distance.

Since all geometric derivatives are computed at the mesh on the reference surface, the following type of expressions are used when evaluating the strains:

$$\frac{\partial \mathbf{r}}{\partial \xi^\alpha} = \frac{\partial \mathbf{r}_\mathrm{R}}{\partial \xi^\alpha} + \zeta_\mathrm{o}\frac{\partial \mathbf{n}}{\partial \xi^\alpha}$$

The degrees of freedom are located on the reference surface.

All loads and boundary conditions are assumed to be applied at the midsurface, so a force acting in the plane of the shell does not cause any bending action if there is an offset.

The numerical integration of the element is performed over the reference surface. If the shell is curved, the area of the actual midsurface and the reference surface differ. This is compensated for by multiplying the weak expressions with an area scale factor, 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.

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

$$|\boldsymbol{\theta}| = \mathrm{acos}(1 + \mathbf{n} \cdot \mathbf{a})$$

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 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

$$\mathbf{n}_j \cdot \mathbf{a}_k + \mathbf{a}_j \cdot \mathbf{n}_k + \mathbf{a}_j \cdot \mathbf{a}_k = 0 \qquad (5\text{-}2)$$

where the values or 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

$$\int_V f(\xi^1,\xi^2,\xi^3)dV = \int_A \int_{-d/2}^{d/2} f_1(\xi^1,\xi^2)f_2(\xi^3)d\varsigma dA = \int_A f_1(\xi^1,\xi^2)F_2(d)dA$$

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

$$\mathbf{a}_0 = \Omega_0 \times \mathbf{n}$$

where Ω_0 is the vector of prescribed nodal rotations. This relation is fully defined only when all three components of Ω_0 are given. It is also possible to prescribe only one or two of the components of Ω_0, while leaving the remaining components free. Because it has no relevance to prescribe the rotation about the normal direction of the shell, it is only possible prescribe individual rotations in a shell local system. In this case, the result becomes one or two constraint relations between the components of $\mathbf{a}_0$. The directions are the edge local coordinate system where $\mathbf{t}_1$ is the tangent to the edge and

$\mathbf{t}_2$ is perpendicular and inward from the edge, in the plane of the shell. These constraints are formulated as

$$\mathbf{t}_2 \cdot \mathbf{a}_0 = -\Omega_{01}$$

$$\mathbf{t}_1 \cdot \mathbf{a}_0 = \Omega_{02}$$

Here Ω_{0i} is the prescribed rotation around the axis $\mathbf{t}_i$.

In a geometrically nonlinear analysis, it is not possible to prescribe individual elements of the rotation vector. If only one or two components have been specified, the remaining components are set to zero. The actual degrees of freedom are then computed as

$$\mathbf{a}_0 = (\mathbf{R}(\Omega_0) - \mathbf{I})\mathbf{n}$$

where $\mathbf{R}$ is a standard rotation matrix, representing the finite rotation about the given rotation vector.

Initial velocities are always given using an angular velocity vector ω as

$$\dot{\mathbf{a}} = \omega \times (\mathbf{n} + \mathbf{a}_0)$$

SYMMETRY AND ANTISYMMETRY BOUNDARY CONDITIONS

It is possible to prescribe symmetry and antisymmetry boundary conditions. As a default, they are expressed in a shell local coordinate system. If applied to a boundary, the normal to the shell is assumed to be the normal to the symmetry or antisymmetry plane. The conditions are

$$\mathbf{u} \cdot \mathbf{n} = 0$$

$$\mathbf{a} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{a} \cdot \mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u} \cdot \mathbf{t}_1 = 0$$

$$\mathbf{u} \cdot \mathbf{t}_2 = 0$$

for the antisymmetry case. Here $\mathbf{t}_1$ and $\mathbf{t}_2$ are two perpendicular directions in the plane of the shell.

When applied to an edge, there is a local coordinate system where $\mathbf{t}_1$ is the tangent to the edge, and $\mathbf{t}_2$ is perpendicular and in the plane of the shell. The assumption is then that $\mathbf{t}_2$ is the normal to the symmetry or antisymmetry plane. The constraints are

$$\mathbf{u}\cdot\mathbf{t}_2 = 0$$

$$\mathbf{a}\cdot\mathbf{t}_2 = 0$$

for the symmetry case and

$$\mathbf{u}\cdot\mathbf{n} = 0$$

$$\mathbf{u}\cdot\mathbf{t}_1 = 0$$

$$\mathbf{a}\cdot\mathbf{t}_1 = 0$$

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.

EXTERNAL LOADS

Contributions to the virtual work from the external load is of the form

$$\mathbf{u}_{\text{test}} \cdot \mathbf{F} + \mathbf{a}_{\text{test}} \cdot [\mathbf{M} \times (\mathbf{n} + \mathbf{a})]$$

where the forces ($\mathbf{F}$) and moments ($\mathbf{M}$) can be distributed over a boundary or an edge or concentrated in a point. The contribution from the normal vector displacement $\mathbf{a}$ is only included in a geometrically nonlinear analysis. Loads are always referred to the midsurface of the element. In the special case of a follower load, defined by its pressure p, the force intensity is

$$\mathbf{F} = -p(\mathbf{n} + \mathbf{a})$$

For a follower load, the change in midsurface area is not taken into account, in order to be consistent with the assumption that thickness changes are ignored.

STRESS AND STRAIN CALCULATIONS

The strains calculated in the element are, as described above, the covariant tensor components. They have little significance for the user, and are internally transformed to a Cartesian coordinate system. This system can be global or element local. The stresses are computed by applying the constitutive law to the thus computed strain tensor.

Each part of the covariant strain ($\gamma_{\alpha\beta}$, $\chi_{\alpha\beta}$, ζ_α) is transformed separately. They correspond to membrane, bending, and shear action, respectively, and it is thus possible to separate the stresses from each of these actions. The membrane stress is defined as

$$\sigma_{\text{m}} = D[\gamma - \gamma_{\text{i}} - \alpha(T_{\text{m}} - T_{\text{ref}}) - \beta_h(c_{\text{m}} - c_{\text{mo,ref}})] + \frac{N_{\text{i}}}{d}$$

where D is the plane stress constitutive matrix, N_{i} are the initial membrane forces, and γ_{i} the initial membrane strains. The influence of thermal strains is included through the midsurface temperature T_{m}, and the hygroscopic swelling through the midsurface moisture concentration, c_{m}.The membrane stress can be considered as the stress at the mid-surface, or as the average through the thickness.

The bending stress is defined as

$$\sigma_{\text{b}} = \frac{Dd}{2}\left[\chi - \chi_{\text{i}} - \alpha\frac{\Delta T}{d} - \beta_h\frac{\Delta c_{\text{mo}}}{d}\right] + \frac{6M_{\text{i}}}{d^2}$$

where χ_{i} is the initial value of the bending part of the strain tensor (actually: the curvature), and M_{i} are the initial bending and twisting moments. ΔT is the temperature difference between the top and bottom surface of the shell, and Δc_{mo} is the difference in moisture concentration between the top and bottom. The bending stress is the stress at the top surface of the shell if no membrane stress is present.

The average transverse shear stress is defined as

$$\sigma_{\mathrm{s}} = \frac{5}{6}\cdot 2G(\zeta-\zeta_{\mathrm{i}})+\frac{Q_{\mathrm{i}}}{d}$$

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

$$\sigma = \sigma_{\mathrm{m}}+z\sigma_{\mathrm{b}}$$

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:

$$\frac{3\sigma_{\mathrm{s}}(1-z^2)}{2}$$

The shell section forces (membrane forces, bending moments, and shear forces) are computed from the stresses as

$$N = d\sigma_{\mathrm{m}}$$

$$M = \frac{d^2}{6}\sigma_{\mathrm{b}}$$

$$Q = d\sigma_{\mathrm{s}}$$

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 $\mathbf{e}_{\mathrm{zl}}$ is the positive normal of the shell surface.

2. The local x direction $\mathbf{e}_{\mathrm{xl}}$ is the projection of the first direction in the material coordinate system ($\mathbf{e}_{\mathrm{x1}}$) on the shell surface

$$\mathbf{e}_{\mathrm{xl}} = \frac{\mathbf{e}_{\mathrm{x1}} - (\mathbf{e}_{\mathrm{x1}} \cdot \mathbf{e}_{\mathrm{zl}})\mathbf{e}_{\mathrm{zl}}}{|\mathbf{e}_{\mathrm{x1}} - (\mathbf{e}_{\mathrm{x1}} \cdot \mathbf{e}_{\mathrm{zl}})\mathbf{e}_{\mathrm{zl}}|}$$

This projection cannot be performed if $\mathbf{e}_{\mathrm{x1}}$ is normal to the shell. In that case the second axis $\mathbf{e}_{\mathrm{x2}}$ of the material system instead defines $\mathbf{e}_{\mathrm{xl}}$ using the same procedure. Thus, if

$$\mathbf{e}_{\mathrm{x1}} \cdot \mathbf{e}_{\mathrm{zl}} > 0.99$$

then

$$\mathbf{e}_{\mathrm{xl}} = \frac{\mathbf{e}_{\mathrm{x2}} - (\mathbf{e}_{\mathrm{x2}} \cdot \mathbf{e}_{\mathrm{zl}})\mathbf{e}_{\mathrm{zl}}}{|\mathbf{e}_{\mathrm{x2}} - (\mathbf{e}_{\mathrm{x2}} \cdot \mathbf{e}_{\mathrm{zl}})\mathbf{e}_{\mathrm{zl}}|}$$

3. At last, the second in-plane direction is generated as

$$\mathbf{e}_{y\mathrm{l}} = \mathbf{e}_{\mathrm{zl}} \times \mathbf{e}_{x\mathrm{l}}$$

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.
- In this section, every reference to stresses is equally valid for strains.

Local edge system

Many features, such as an edge load, allow input in an edge local coordinate system. The orthogonal local edge coordinate system directions $\mathbf{x}_{l,}$ $\mathbf{y}_l$, and $\mathbf{z}_l$ are defined so that:

- The first direction ($\mathbf{x}_l$) is along the edge. This direction can be visualized by selecting the **Show edge directions arrows** check box in the **View** settings.
- The third direction ($\mathbf{z}_l$) is the same as the shell normal direction. The shell normal direction can be visualized in the default plot named `Undeformed geometry`, or by adding a **Coordinate System Surface** plot and selecting the **Shell, Local System**.
- The second direction ($\mathbf{y}_l$) is in the plane of the shell and orthogonal to the edge. It is formed by the cross product of $\mathbf{z}_l$ and $\mathbf{x}_l$; $\mathbf{y}_l = \mathbf{z}_l \times \mathbf{x}_l$.

When an edge is shared between two or more boundaries, the directions may not always be unique. It is then possible to use the control **Face Defining the Local Edge System** to select from which boundary the normal direction $\mathbf{z}_l$ should be picked. The default is **Use face with lowest number**. If the geometry selection contains several edges, the only available option is **Use face with lowest number**, since the list of adjacent boundaries would then be different for each edge.

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 for example 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.

Rigid Domain for Shells

The inertial properties mass (m) and moment of inertia tensor (I) of a rigid shell takes the finite thickness into account. They are computed as:

$$m = \int_V \rho dV = \int_A \rho d\ dA$$

$$\mathbf{X}_{\mathrm{M}} = \frac{1}{m}\int_V \rho \mathbf{X} dV = \frac{1}{m}\int_V \rho(\mathbf{r} + \varsigma\mathbf{n})dV = \frac{1}{m}\int_A \rho d\mathbf{r} dA$$

$$I = \int_V (((\mathbf{X}-\mathbf{X}_{\mathrm{M}})^T \cdot (\mathbf{X}-\mathbf{X}_{\mathrm{M}}))E_3 - (\mathbf{X}-\mathbf{X}_{\mathrm{M}}) \cdot (\mathbf{X}-\mathbf{X}_{\mathrm{M}})^T)\rho dV =$$

$$\int_A (((\mathbf{r}-\mathbf{X}_{\mathrm{M}})^T \cdot (\mathbf{r}-\mathbf{X}_{\mathrm{M}}))E_3 - (\mathbf{r}-\mathbf{X}_{\mathrm{M}}) \cdot (\mathbf{r}-\mathbf{X}_{\mathrm{M}})^T)\rho d\ dA +$$

$$\int_A (E_3 - \mathbf{n}\mathbf{n}^T)\rho\frac{d^3}{12}\ dA$$

where E_3 and $\mathbf{X}_{\mathrm{M}}$ are the identity matrix and the center of mass of the rigid domain, respectively. The last term in I is accounting for the finite thickness, and there the fact that $\mathbf{n}^T \cdot \mathbf{n} = 1$ has been 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 approximative.
- *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.

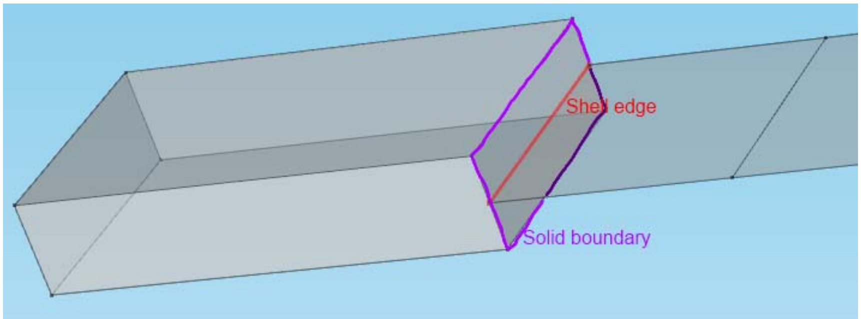


Figure 5-2: An example of a shell extending perpendicular to the solid boundary.

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:

$$u'_{sld}(y',z') = u'_{sh}(y',0) + z'a'_x(y',0)$$
$$v'_{sld}(y',z') = v'_{sh}(y',0) + z'a'_y(y',0)$$

where the subscripts sld and sh represent ‘solid’ and ‘shell’, respectively, and $\mathbf{a}'$ is the displacement of the shell normal, represented in the local directions.

The values on the solid boundaries should be interpreted as mapped using an extrusion operator from the shell edge.

A simple connection for the transverse direction can be generated by

$$w'_{sld}(y',z') = w'_{sh}(y',0)$$

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

$$\varepsilon_{z'} = -\frac{\upsilon}{E}(\sigma_{x'} + \sigma_{y'}) = K_1(x',y') + 2K_2(x',y')z'$$

The transverse displacement is then, after integration with respect to z':

$$w' = K_0(x',y') + K_1(x',y')z' + K_2(x',y')z'^2$$

Thus, in order to obtain a stress-free transverse displacement, this deformation must be allowed. Note that the last term makes the solid thinner on one side of the midsurface and thicker on the other. The constant term—that is, the displacement at the midsurface of the solid—is known from the shell midsurface displacement. The other two terms depend on the stress state and thus on derivatives of the displacement, which are not readily available. The K_1 term is caused by the membrane action and the K_2 term by the bending action.

For the transverse shear stress, Hooke's law gives

$$\gamma_{x'z'} = \frac{\partial u'}{\partial z'} + \frac{\partial w'}{\partial x'} = \frac{\tau_{x'z'}}{G} = \frac{C_1}{G}(1 - z^2)$$

which can be reformulated as

$$\frac{\partial u'}{\partial z'} = \frac{C_1}{G}(1 - z^2) - \frac{\partial w'}{\partial x'} = \frac{C_1}{G}(1 - z^2) - \frac{\partial K_0}{\partial x'} - \frac{\partial K_1}{\partial x'}z' - \frac{\partial K_2}{\partial x'}z'^2$$

Note that as the K_1 term is related to membrane action, it cannot contribute to the transverse shear stress. Since the derivative in the x' direction cannot be controlled by a condition on the boundary, it is necessary to make an assumption about $u'(z')$. An integration with respect to z' gives

$$u' = M_0(x',y') + M_1(x',y')z' + M_3(x',y')z'^3$$

This shows that a third power of z' is required in order to be able to represent the correct shear strain contribution.

It is, however, not possible to directly determine the coefficients in front of the additional terms, since they depend on the actual stress state. The idea is here to introduce them as dependent variables in the problem. These variables are defined by extra shape functions on the shell edge.

The constraints applied on the solid can then be written as

$$\begin{aligned}u'_{\mathrm{sld}}(y',z') &= u'_{\mathrm{sh}}(y',0) + z'a'_x(y',0) + q_{\mathrm{u}}(y',0)(z^3 - k_{\mathrm{u}}z)d\\ v'_{\mathrm{sld}}(y',z') &= v'_{\mathrm{sh}}(y',0) + z'a'_y(y',0)\\ w'_{\mathrm{sld}}(y',z') &= w'_{\mathrm{sh}}(y',0) + q_{\mathrm{w1}}(y',0)zd + q_{\mathrm{w2}}(y',0)(z^2 - k_{\mathrm{w2}})d\end{aligned}$$

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 $k_{\mathrm{u}} = 3/5$ and $k_{\mathrm{w2}} = 1/5$ are explained below. The variable q_{w1} is proportional to the membrane axial strain, the variable q_{w2} is proportional to the bending strain, and the variable q_{u} is proportional to the transverse shear strain. Since these 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 $\sigma_{\mathrm{x'}}$ and the reaction force for w' is the traction $\tau_{\mathrm{x'z'}}$. When taking the variation of the new dependent variables, these enforce the following constraints:

$$test(q_{\mathrm{u}})\int_{\frac{-d}{2}}^{\frac{d}{2}}(z^3-k_{\mathrm{u}}z)\sigma_{x'}dz' = 0$$

$$test(q_{\mathrm{w1}})\int_{\frac{-d}{2}}^{\frac{d}{2}}z\tau_{x'z'}dz' = 0$$

$$test(q_{\mathrm{w2}})\int_{\frac{-d}{2}}^{\frac{d}{2}}(z^2-k_{\mathrm{w2}})\tau_{x'z'}dz' = 0$$

The equation with q_{w1} is trivially fulfilled because the shear stress is an even function of z'. Inserting the known stress distributions gives equations that can be solved for $\mathbf{k}_{\mathrm{u}}$ and $\mathbf{k}_{\mathrm{w2}}$.

The constraint expressions must now be formulated in global directions. As a start, the constraints are written on vector form in local directions as

$$\mathbf{u'}_{\mathrm{sld}} = \mathbf{u'}_{\mathrm{sh}} + z'\mathbf{a}' + \mathbf{q} \qquad (5\text{-}3)$$

where

$$\mathbf{q} = \begin{bmatrix} q_{\mathrm{u}}(y',0)(z^3-k_{\mathrm{u}}z) \\ 0 \\ q_{\mathrm{w1}}(y',0)z + q_{\mathrm{w2}}(y',0)(z^2-k_{\mathrm{w2}}) \end{bmatrix}$$

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

$$z' = (\mathbf{X}_{\mathrm{sld}} - \mathbf{X}_{\mathrm{sh}})\cdot\mathbf{N}_{\mathrm{sh}}$$

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.

$$z' = (\mathbf{x}_{\mathrm{sld}} - \mathbf{x}_{\mathrm{sh}}) \cdot \mathbf{n}_{\mathrm{sh}}$$

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 $\mathbf{\Phi}$ be the matrix that transforms displacements from the global system to the local system:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{e}_{x'} & \mathbf{e}_{y'} & \mathbf{e}_{z'} \end{bmatrix}$$

The expression for the constraints in global directions then becomes

$$\mathbf{u}_{\mathrm{sld}} = \mathbf{\Phi}^T \mathbf{u}'_{\mathrm{sld}} = \mathbf{\Phi}^T (\mathbf{u}'_{\mathrm{sh}} + z'\mathbf{a}' + \mathbf{q}) = \mathbf{u}_{\mathrm{sh}} + z'\mathbf{a} + \mathbf{\Phi}^T \mathbf{q}$$

The only transformation actually needed is thus the projection of the **q** vector. For a linear case, the transformation can be written as

$$\mathbf{\Phi} = \begin{bmatrix} s\mathbf{t}_1 \times \mathbf{N}_{\mathrm{sh}} & s\mathbf{t}_1 & \mathbf{N}_{\mathrm{sh}} \end{bmatrix}$$

where **N** is the undeformed shell normal (`shell.an`) and $\mathbf{t}_1$ is the shell edge tangent (`shell.tle`). The coefficient s is either 1 or −1, and is selected so that the x' direction coincides with the outward normal of the solid.

For a geometrically nonlinear case, the corresponding deformed directions are used.

When an offset is used for the shell, it is assumed that the center of the connection is at the actual shell midsurface.

Beam Perpendicular to Solid

This is the analogous case in 2D. Beam theory assumes that the stress in the out-of-plane direction is zero. It is thus only physically sound to connect to a model where the plane stress assumption is used. The derivation above still remains valid with the following exceptions: the displacements in the local y' direction are zero and the tangent direction $\mathbf{t}_1$ is replaced by the out-of-plane direction.

The shell thickness is replaced by twice the effective radius of the beam in the equations defining the displacements.

Shell Parallel to Solid

The case where the shell is parallel to the boundary of the solid exists in two versions - *Shared* and *Parallel*.

In the shared case, the shell is modeled on a boundary which is a face of the solid. In this case, it assumed that the names of the displacement degrees of freedom in the solid and shell interfaces are not the same. If the same names are used, there is no need to use a connection feature, since the coupling is then automatic. A shell offset can be used to model an actual distance between the boundaries. For a layer 'glued' on the solid, the offset would equal half the shell thickness.

In the parallel case, a separate boundary is used for modeling the shell. The distance between the shell and the face of the solid is taken into account when setting up the constraints, so that

$$\mathbf{u}_{\mathrm{sld}} = \mathbf{u}_{\mathrm{shl}} - \zeta\mathbf{a}$$

where ζ is a distance from the solid to the shell. The right hand side is mapped from the shell to the solid using an extrusion operator. The default is that ζ is half the shell thickness, but you can also use the geometrical distance between the boundaries, or a user defined distance.

CONNECTION BETWEEN SHELLS AND BEAMS

When connecting elements from the Shell interface with elements from the Beam interface, the following must be noted:

- The rotational degrees of freedom have different definitions in the two interfaces. The Beam interface uses a rotation vector representation, whereas the Shell interface uses the normal vector displacements.
- In general, a shell element locally has three translational, but only two rotational, degrees of freedom. The rotation around the normal is not part of the shell theory and is constrained to zero. The corresponding rotational degree of freedom in a beam should 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 **hell 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 therms of the degrees of freedom in the shell as

$$\mathbf{u}_\mathrm{b} = \mathbf{u}_\mathrm{s} + (\mathbf{d}_0 \cdot \mathbf{n})\mathbf{a}$$

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

$$\boldsymbol{\theta}_\mathrm{b} = \mathbf{n} \times \mathbf{a}$$

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

$$(\boldsymbol{\theta}_\mathrm{b} - \mathbf{n} \times \mathbf{a}) \cdot \mathbf{t}_i \qquad i = 1, 2$$

which can be simplified to

$$\begin{aligned} \boldsymbol{\theta}_\mathrm{b} \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 = 0 \\ \boldsymbol{\theta}_\mathrm{b} \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 = 0 \end{aligned}$$

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

$$\mathbf{u}_\mathrm{s} = \mathbf{u}_\mathrm{b} + \boldsymbol{\theta}_\mathrm{b} \times (\mathbf{X}_\mathrm{s} - \mathbf{X}_\mathrm{b})$$

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

$$\begin{aligned}\boldsymbol{\theta}_\mathrm{b} \cdot \mathbf{t}_1 + \mathbf{a} \cdot \mathbf{t}_2 &= 0\\ \boldsymbol{\theta}_\mathrm{b} \cdot \mathbf{t}_2 - \mathbf{a} \cdot \mathbf{t}_1 &= 0\end{aligned}$$

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.

Reference for the Shell Interface

1. D. Chapelle and K.J. Bathe, *The Finite Element Analysis of Shells—Fundamentals*, Springer-Verlag, Berlin Heidelberg, 2003.

The Shell and Plate Interfaces

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 **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. Geometric nonlinearity can be taken into account.

The **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 Linear Elastic Material is the only available material model. It adds a linear elastic equation for the displacements and has a Settings window to define the elastic material properties.

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 **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`.

THICKNESS

Define the **Thickness** d by entering a value or expression in the field. The default is 0.01 m. Use the Change Thickness node to define a different thickness in parts of the shell or plate. The thickness can be variable if an expression is used.

Offset Definition

This section is available for the Shell interface only. For theory, see Offset.

If the actual shell midsurface is not the boundary on which the mesh exists, it is possible to prescribe an offset in the direction of the surface normal by using an offset definition. The offset is defined as positive if the shell midsurface is displaced from the meshed boundary in the direction of the positive shell normal.

Select an option from the **Offset definition** list—**No offset** (the default), **Physical offset**, or **Relative offset**. The default **No offset** means that the modeled boundary coincides with the shell midsurface.

- 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.

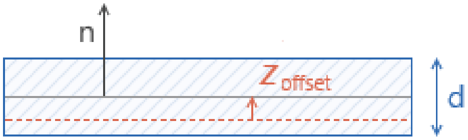


Figure 5-3: Meshed boundary indicated in red. The vector **n** *indicates the positive orientation of the shell normal.*

- 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.

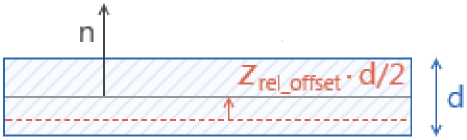


Figure 5-4: Meshed boundary indicated in red.The vector **n** *indicates the positive orientation of the shell normal.*

Use the Change Thickness node to define a different offset in parts of the shell.

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.

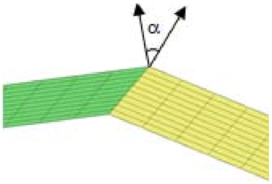
Reference Point for Moment Computation

Enter the default coordinates for the **Reference point for moment computation** $\mathbf{x}_{\mathrm{ref}}$. 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.



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.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **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_Models/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):

- Antisymmetry
- Attachment
- Beam Connection
- Body Load
- Change Thickness
- Edge Load
- External Stress
- Face Load
- Hygroscopic Swelling
- Initial Values
- Initial Stress and Strain
- Linear Elastic Material
- No Rotation
- Periodic Condition
- Phase
- Pinned
- Point Load
- Point Mass
- Point Mass Damping
- Prescribed Acceleration
- Prescribed Displacement/Rotation
- Prescribed Velocity
- Rigid Connector
- Shell Local System
- Simply Supported
- Solid Connection
- Symmetry
- Thermal Expansion (for constraints)
- Thermal Expansion (for materials)
- Thermal Expansion (Attachment)
- Thermal Expansion (Rigid Connector)

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:

- Added Mass
- Applied Force (Rigid Connector)
- Applied Moment (Rigid Connector)
- Damping
- Free
- Mass and Moment of Inertia (Rigid Connector)
- Fixed Constraint[2]
- Gravity[1]
- Predeformation
- Rigid Domain
- Rotating Frame[1]
- Spring Foundation

[1] This is selected from the **Face and Volume Loads** submenu for this interface.

[2] This is selected from the **Face Constraints** 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** $\frac{\partial u}{\partial t}$

- **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

Change Thickness

Use the **Change Thickness** node to model regions with a thickness or offset other than the overall values defined in the interface **Thickness** section.

CHANGE THICKNESS

Enter a value for the **Thickness** d. This value replaces the overall thickness for the boundaries selected above. The thickness can be variable if an expression is used.

Offset Definition

If the actual shell midsurface is not the boundary on which the mesh exists, it is possible to prescribe an offset in the direction of the surface normal by using an offset definition. The offset is defined as positive if the shell midsurface is displaced from the meshed boundary in the direction of the positive shell normal.

Select an option from the **Offset definition** list—**From parent** (the default), **No offset, Physical offset**, or **Relative offset**. The default **From parent** means that the offset is as defined in the **Thickness** section of the shell interface.

- For **No offset** it means that the modeled boundary coincides with the shell midsurface.

- 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.

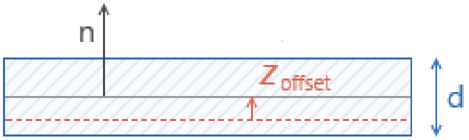


Figure 5-5: Meshed boundary indicated in red. The vector **n** *indicates the positive orientation of the shell normal.*

- 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.]

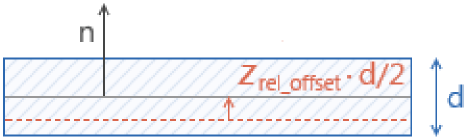


Figure 5-6: Meshed boundary indicated in red. The vector **n** *indicates the positive orientation of the shell normal.*

For theory, see Offset.

LOCATION IN USER INTERFACE

Context Menus

Shell>Change Thickness

Plate>Change Thickness

Ribbon

Physics tab with **Shell** selected:

Boundaries>Shell>Change Thickness

Physics tab with **Plate** selected:

Domains>Plate>Change Thickness

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

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

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 the **Shell Local System** node added under **Definitions**.

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

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

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.

MODEL INPUTS

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. Iff needed, you can add a through-thickness temperature gradient in the **Thermal Bending** section.

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 non-isotropic coefficient of thermal expansion is used, the axis orientations are given by the coordinate system selection in the parent node.

Enter a value or expression of the **Strain reference temperature** T_{ref}, which is the reference temperature where the thermal strain is zero.

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.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Thermal Expansion

Plate>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

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the thickness direction of the shell. 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 mid-surface 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` m^3/kg. When a non-isotropic 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.

LOCATION IN USER INTERFACE

Context Menus

Shell>Linear Elastic Material>Hygroscopic Swelling

Plate>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.

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_{i}
- **Initial moment** M_{i}
- **Initial out-of-plane shear force** Q_{i}

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_i
- **Initial bending strain** χ_i
- **Initial transverse shear strain** ζ_i

- 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

Plate>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.

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** $\sigma_{\mathrm{ext,m}}$ ($\sigma_{\mathrm{ext,b}}$, $\sigma_{\mathrm{ext,s}}$) 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

Plate>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, 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**.

FACE DEFINING THE LOCAL EDGE SYSTEM

This setting is used in conjunction with a **Local edge system**. If displacement or rotations 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

Specify the coordinate system to use for specifying the prescribed displacement/rotation. The coordinate system selection is based on the geometric entity level.

Boundaries (Plate interface: Domains)

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate system

Edges (Plate interface: Boundaries)

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.

Points

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- Any additional user-defined coordinate system

Depending on the selected coordinate system, the displacement and rotation components change accordingly.

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.

- **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

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

THERMAL EXPANSION PROPERTIES

Specify the thermal properties that define the thermal strain. This is a description of the thermal expansion of the urroundings 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 non-isotropic 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 **Strain 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, domain, or point 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 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.

Boundaries

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate system

Edges

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.

Points

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- Any additional user-defined coordinate system

Depending on the selected coordinate system, the velocity components change accordingly.

FACE DEFINING THE LOCAL EDGE SYSTEM

This setting is used in conjunction with a **Local edge 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 and is only visible if the selected coordinate system is **Local edge system**.

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.

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, domain, or point 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.

Boundaries

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate system

Edges

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.

Points

From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- Any additional user-defined coordinate system

Depending on the selected coordinate system, the acceleration components change accordingly.

FACE DEFINING THE LOCAL EDGE SYSTEM

This setting is used in conjunction with a **Local edge 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 and is only visible if the selected coordinate system is **Local edge system**.

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.

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.

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.

FACE DEFINING THE LOCAL EDGE SYSTEM (EDGES ONLY)

This setting is used in conjunction with a **Local edge 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.

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.

SYMMETRY (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**.

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

Antisymmetry

The **Antisymmetry** node adds an edge or boundary condition that defines an antisymmetry edge or boundary.

FACE DEFINING THE NORMAL DIRECTION (EDGES ONLY)

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**.

In the *COMSOL Multiphysics Reference Manual:*

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Symmetry and Antisymmetry Boundary Conditions

You can activate and deactivate this boundary condition by assigning it to a constraint group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell>Face Constraints>Antisymmetry

Shell>More Constraints>Antisymmetry (Edge)

Plate>More Constraints>Antisymmetry (Boundary)

Ribbon

Physics tab with **Shell** selected:

Boundaries>Face Constraints>Antisymmetry

Edges>More Constraints>Antisymmetry

Physics tab with **Plate** selected:

Boundaries>More Constraints>Antisymmetry

Body Load

Add a **Body Load** to boundaries (for the Plate interface add it to domains). The loads are defined in the given coordinate system.

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	$\mathbf{F}_A$	N/m^2	boundaries domains	3D (x, y, z) 2D (x, y, z)

Total force	$\mathbf{F}_{tot}$	N	boundaries
			domains
				3D(x, y, z)
				2D(x, y, z)

Pressure	p	Pa	boundaries domains	3D 2D

- A positive pressure is directed in the negative element normal direction.
- The pressure load is a ‘follower load’. The direction changes with deformation in a geometrically nonlinear analysis.
- After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining with another physics interface, it is also possible to choose a predefined load from this list.

MOMENT

Enter values or expressions for the components of the moment face load $\mathbf{M}_A$.

- You can add the Phase subnode to specify the phase of this load in a frequency domain analysis.
- You can specify this load to be a Harmonic Perturbation in a frequency domain analysis.
- You can assign this load to a load group. See Load Cases in the Structural Mechanics Modeling chapter.

LOCATION IN USER INTERFACE

Context Menus

Shell>Face and Volume Loads>Face Load

Plate>Face and Volume Loads>Face Load

Membrane>Face and Volume Loads>Face Load

Ribbon

Physics tab with **Shell** or **Membrane** selected:

Boundaries>Face and Volume Loads>Face Load

Physics tab with **Plate** selected:

Domains>Face and Volume Loads>Face Load

Edge Load

Add an **Edge Load** as a force or moment distributed along an edge (for the Plate interface add it to boundaries). The load is defined in the given local coordinate system.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the load. From the **Coordinate system** list select from:

- **Global coordinate system** (the standard global coordinate system).

- **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 LOCAL EDGE SYSTEM

This setting is used in conjunction with a **Local edge system**. When the load is applied to 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 and is only visible if the selected coordinate system is **Local edge 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	$\mathbf{F}_L$	N/m	edges	3D
			boundaries	2D
Force per unit area	$\mathbf{F}_A$	N/m^2	edges	3D
			boundaries	2D
Total force	$\mathbf{F}_{tot}$	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	$\mathbf{M}_L$	N	edges	3D
			boundaries	2D
Moment per unit area	$\mathbf{M}_A$	$\mathrm{N{\cdot}m/m^2}$	edges	3D
			boundaries	2D
Total moment	$\mathbf{M}_{\mathrm{tot}}$	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.

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

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; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = \mathbf{u}(\mathbf{x}_\mathrm{s})$ and $\mathbf{a}(\mathbf{x}_\mathrm{d}) = \mathbf{a}(\mathbf{x}_\mathrm{s})$. 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; $\mathbf{u}(\mathbf{x}_\mathrm{d}) = -\mathbf{u}(\mathbf{x}_\mathrm{s})$ and $\mathbf{a}(\mathbf{x}_\mathrm{d}) = -\mathbf{a}(\mathbf{x}_\mathrm{s})$. 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** $\mathbf{k}_\mathrm{F}$. 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, 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.

If any point on the edges having the periodic condition is located on the axis of cyclic symmetry, enter the **Axis direction** vector, $\mathbf{t}_\mathrm{c}$. 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 $u_i(\mathbf{x}_\mathrm{d}) = u_i(\mathbf{x}_\mathrm{s})$ or $u_i(\mathbf{x}_\mathrm{d}) = -u_i(\mathbf{x}_\mathrm{s})$ respectively. Each selected rotation component will be connected by $a_i(\mathbf{x}_\mathrm{d}) = a_i(\mathbf{x}_\mathrm{s})$ or $a_i(\mathbf{x}_\mathrm{d}) = -a_i(\mathbf{x}_\mathrm{s})$ 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*.

Vibrations of an Impeller: Application Library path **Structural_Mechanics_Module/Dynamics_and_Vibration/impeller**

LOCATION IN USER INTERFACE

Context Menus

Shell>Connections>Periodic Condition

Plate>Periodic Condition

Shell>Connections>Periodic Condition>Destination Selection

Plate>Periodic Condition>Destination Selection

Ribbon

Physics tab with **Shell** selected:

Edges>Connections>Periodic Condition

Physics tab with **Plate** selected:

Boundaries>Periodic Condition

Physics tab with **Periodic Condition** node selected in the model tree:

Attributes>Destination Selection

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 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 the 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 values. 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.

$$\mathbf{X}_\mathrm{C} = \mathbf{X}_{\mathrm{C,input}} + \mathbf{X}_\mathrm{offset}$$

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

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

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 non-isotropic 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 **Strain 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 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 Attachments is included 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

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

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 non-isotropic 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 **Strain 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:

$$F_{\mathrm{freq}} = F(f)\cdot\cos(2\pi ft+\phi)$$

PHASE

Add the phase angle $\mathbf{F}_{\mathrm{ph}}$ 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 $\mathbf{M}_{\mathrm{ph}}$ 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 non-zero 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

6

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.

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.

The Right Cauchy-Green tensor $\mathbf{C}$ is generally defined as

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

Since the tangential deformation gradient does not contain any information about the transversal stretch λ_n, it must be augmented when creating $\mathbf{C}$.

$$\mathbf{C} = \mathbf{F}_T^t\mathbf{F}_T + \lambda_\mathrm{n}^2\mathbf{N} \cdot \mathbf{N}^t = \mathbf{F}_T^t\mathbf{F}_T + (1 + u_\mathrm{nn})^2\mathbf{N} \cdot \mathbf{N}^t$$

Similar augmentation is also needed in the full deformation gradient tensor $\mathbf{F}$.

$$\mathbf{F} = \mathbf{I} + \nabla_T\mathbf{u} + u_\mathrm{nn}\mathbf{N} \cdot \mathbf{N}^t$$

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

$$J_{\mathrm{A}} = \frac{J}{1+u_{\mathrm{nn}}}$$

In the case of geometrically linear analysis, a linearized version of the strain tensor is used.

Constitutive Relation and Weak Expressions

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

$$\mathbf{u}_{\mathrm{test}} \cdot \mathbf{F}$$

where the forces ($\mathbf{F}$) can be distributed over a boundary or an edge or be concentrated in a point. In the special case of a follower load, defined by its pressure p, the force intensity is $\mathbf{F} = -p\mathbf{n}$ where $\mathbf{n}$ is the normal in the deformed configuration.

For a follower load, the change in midsurface area is taken into account, and integration of the load is done in the spatial frame.

Stress Calculations

The stresses are computed by applying the constitutive law to the computed strains.

The membrane does not support transverse and bending forces so the only section forces it support is the membrane force defined as:

$$N = d \cdot \bar{s}$$

where $\bar{s}$ is the local stress tensor and contains only in-plane stress components.

The Membrane Interface

The **Membrane (mbrn)** interface ( ), found under the **Structural Mechanics** branch ( ) when adding a physics interface, is mainly used to model prestressed membranes, but can also be used to model a thin cladding on a solid. Membranes can be considered as plane stress elements on boundaries in 3D with a possibility to deform both in the in-plane and out-of-plane directions. There is also a version of the membrane interface for 2D axisymmetric problems. The membrane interface is then applicable to lines since that is what represents boundaries.

The difference between a shell and a membrane is that the membrane does not have any bending stiffness. In most applications, the membrane is used by itself and not as a cladding. A tensile prestress is then necessary in order to avoid singularity because a membrane with no stress or compressive stress has no transverse stiffness. To include the prestress effect, you must enable geometric nonlinearity for the study step.

The Linear Elastic Material is the default material, which adds a linear elastic equation for the displacements and has a Settings window to define the elastic material properties. This material model can also be combined with viscoelasticity.

With the Nonlinear Structural Materials Module, you can also model Nonlinear Elastic and Hyperelastic materials, and add options such as Plasticity, Creep, and Viscoplasticity.

For a detailed overview of the functionality available in each product, visit 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 if axial symmetry, an **Axial Symmetry** node is also added. From the **Physics** toolbar, you can then add other nodes that implement, for example, loads and constraints. You can also right-click **Membrane** to select physics features from the context menu.

SETTINGS

The **Label** is the default physics interface name.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern

<name>.<variable_name>. In order to distinguish between variables belonging to different physics interfaces, the name string must be unique. Only letters, numbers and underscores (_) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is mbrn.

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** $\mathbf{x}_{\mathrm{ref}}$. 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

To display this section, click the **Show** button (👁) and select **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
- Antisymmetry
- Body Load[1]
- Creep
- Damping
- Edge Load[3]
- External Stress
- Face Load[3]
- Fixed Constraint
- Free
- Gravity[1]
- Hygroscopic Swelling
- Hyperelastic Material
- Initial Values
- Nonlinear Elastic Material
- Phase
- Plasticity
- Point Load
- Point Load (on Axis)
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement[2]
- Prescribed Velocity
- Ring Load
- Rotating Frame[1]
- Spring Foundation
- Symmetry
- 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 1** is selected by default. The **Coordinate system** list contains any additional boundary coordinate systems that the model includes. The coordinate system is used for interpreting directions of orthotropic and anisotropic material data and when stresses or strains are presented in a local system. Many of the possible subnodes inherit the coordinate system settings.

LINEAR ELASTIC MATERIAL

To use a mixed formulation by adding the pressure as an extra dependent variable to solve for, select the **Nearly incompressible material** check box. For a material with a very low compressibility, using only displacements as degrees of freedom may lead to a numerically ill-posed problem.

Define the **Solid model** and the linear elastic material properties.

Solid Model

To use a mixed formulation by adding the pressure as an extra dependent variable to solve for, select the **Nearly incompressible material** check box.

Select a linear elastic **Solid model**—**Isotropic** (the default), **Orthotropic**, or **Anisotropic**. Select:

- **Isotropic** for a linear elastic material that has the same properties in all directions.
- **Orthotropic** for a linear elastic material that has different material properties in orthogonal directions, so that its stiffness depends on the properties E_i, 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** $\mathbf{E}$, **Poisson's ratio** ν, and the **Shear modulus** $\mathbf{G}$. This defines the relationship between engineering shear strain and shear stress. It is applicable only to an *orthotropic material* and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}$$

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** $\mathrm{S}_{\mathrm{ext}}$ as **Isotropic**, **Diagonal**, or **Symmetric** depending on the properties of the tensor. The tensor components are interpreted in the selected coordinate system. If a stress tensor announced by a physics interface is selected, the coordinate system setting is ignored — the orientation is handled internally. Choose a **Contribution type**—**Add to stress tensor** 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

7

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 is described in this section:

Beam Cross Section

About Beams

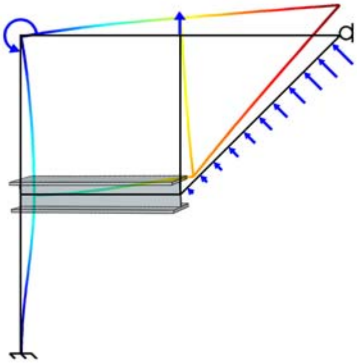
A *beam* is a slender structure that can be fully described by its cross section properties such as area, moments of inertia, and torsional constant. Beams are the choice for modeling reinforcements in 3D solids and shell structures, as well as in 2D solids under the plane stress assumption. Naturally,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



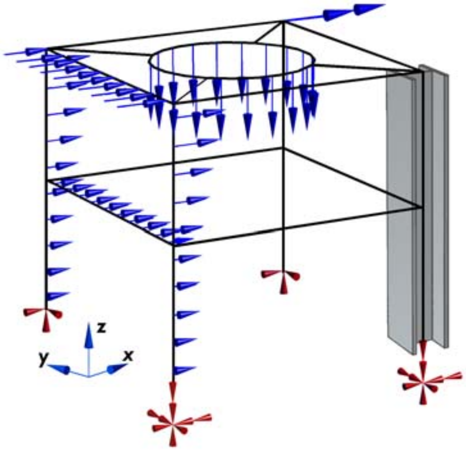
In-plane beams are defined on edges in 2D. They can be used separately or as stiffeners to 2D solid elements.

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.

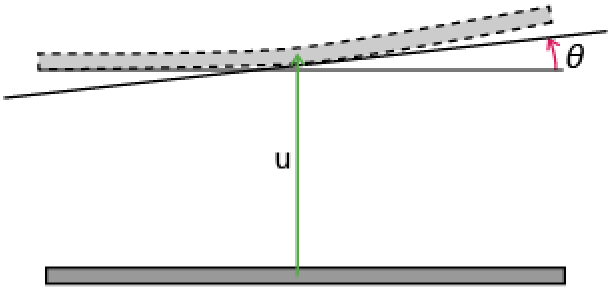


Figure 7-1: The degrees of freedom in the Beam interface,

Geometric Variables

The beam formulations are based on that the 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 7-1 the important geometric variables used in the Beam interface are summarized.

TABLE 7-1: GEOMETRIC VARIABLE 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 7-1: GEOMETRIC VARIABLE 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.muz`	Max shear stress factor in local z-direction
κ_{z}	`beam.kappaz`	Shear correction factor along local z-axis
r_{gz}	`beam.rgz`	Radius of gyration, local z-direction
J	`beam.J_beam`	Torsional constant
W_{t}	`beam.Wt`	Torsional section modulus
r_{e}	`beam.re`	Equivalent radius
	`beam.yBeam1` `beam.zBeam1`	Stress evaluation point 1, local coordinates
	`beam.yBeam2` `beam.zBeam2`	Stress evaluation point 2, local coordinates
	`beam.yBeam3` `beam.zBeam3`	Stress evaluation point 3, local coordinates
	`beam.yBeam4` `beam.zBeam4`	Stress evaluation point 4, local coordinates

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, $\boldsymbol{\theta}$, are interpolated as

$$\begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\theta} \end{Bmatrix} = [\mathbf{N}]\begin{Bmatrix} \mathbf{u}_1 \\ \boldsymbol{\theta}_1 \\ \mathbf{u}_2 \\ \boldsymbol{\theta}_2 \end{Bmatrix}$$

where the subscript refers to the two nodes of the element, and **N** is a matrix of shape functions.

$$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & N_5 & 0 & N_4 & 0 & 0 & 0 & N_6 \\ 0 & 0 & N_3 & 0 & -N_5 & 0 & 0 & 0 & N_4 & 0 & -N_6 & 0 \\ 0 & 0 & 0 & M_1 & 0 & 0 & 0 & 0 & 0 & M_2 & 0 & 0 \\ 0 & 0 & M_3 & 0 & M_5 & 0 & 0 & 0 & M_4 & 0 & M_6 & 0 \\ 0 & -M_3 & 0 & 0 & 0 & M_5 & 0 & -M_4 & 0 & 0 & 0 & M_6 \end{bmatrix}$$

The shape functions for the Euler-Bernoulli case are expressed in the local coordinate ξ, ranging from 0 to 1, as

$$\begin{array}{llll} N_1 = 1-\xi & N_2 = \xi & & \\ N_3 = 1-3\xi^2+2\xi^3 & N_4 = 3\xi^2-2\xi^3 & N_5 = L(\xi-2\xi^2+\xi^3) & N_6 = L(-\xi^2+\xi^3) \\ M_1 = 1-\xi & M_2 = \xi & & \\ M_3 = -\dfrac{6}{L}(\xi-\xi^2) & M_4 = \dfrac{6}{L}(\xi-\xi^2) & M_5 = 1-4\xi+3\xi^2 & M_6 = -2\xi+3\xi^2 \end{array}$$

where L is the length of the beam element.

For the Timoshenko case, the shape functions are modified, so that they depend on the degree of shear flexibility. Define

$$\Phi_2 = \frac{12EI_{zz}}{G\kappa_y AL^2} = \frac{24(1+\nu)I_{zz}}{\kappa_y AL^2}$$

$$\Phi_3 = \frac{12EI_{yy}}{G\kappa_z AL^2} = \frac{24(1+\nu)I_{yy}}{\kappa_z AL^2}$$

which represent the ratios between bending and shear stiffness in the two principal directions. The shape functions are then modified so that

$$N_3^i = \frac{(N_3 + \Phi_i \hat{N}_3)}{1+\Phi_i} \qquad N_4^i = \frac{(N_4 + \Phi_i \hat{N}_4)}{1+\Phi_i}$$

$$N_5^i = \frac{(N_5 + \Phi_i \hat{N}_5)}{1+\Phi_i} \qquad N_6^i = \frac{(N_6 + \Phi_i \hat{N}_6)}{1+\Phi_i}$$

$$M_5^i = \frac{(M_5 + \Phi_i \hat{M}_5)}{1+\Phi_i} \qquad M_6^i = \frac{(M_6 + \Phi_i \hat{M}_6)}{1+\Phi_i}$$

where

$$\hat{N}_3 = 1-\xi \qquad \hat{N}_4 = \xi$$

$$\hat{N}_5 = \frac{L}{2}(\xi-\xi^2) \quad \hat{N}_6 = -\frac{L}{2}(\xi-\xi^2)$$

$$\hat{M}_5 = 1-\xi \qquad \hat{M}_6 = \xi$$

The superscript i, indicates that the shape functions for bending are no longer the same in the two principal directions. The shape functions with $i = 2$ are used for bending in the local y direction, and the shape functions with $i = 3$ are used for bending in the local z direction.

Geometric Nonlinearity

You can use the beam interface for modeling problems with large displacements and rotations, but small strains. A so called co-rotational formulation is used. The displacement of each individual beam element is decomposed into a rigid body translation and rotation, and a local response of the rotated element which is linear.

The assumption that the individual element behaves linearly implies that you must use a fine mesh if the curvature of the deformed beam is large. The difference in rotation between the end points 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, $\boldsymbol{\theta}$. The rotation at the midpoint is approximated as the arithmetic mean of the rotations at the nodes,

$$\boldsymbol{\theta}_{\mathrm{M}} = \frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2}{2}$$

The rigid rotation is then represented by a rotation matrix $\mathbf{R}_{\mathrm{r}}$, corresponding to this midpoint rotation. It is given by

$$\mathbf{R}_{\mathrm{r}} = \mathbf{I} + \frac{\sin|\boldsymbol{\theta}_{\mathrm{M}}|}{|\boldsymbol{\theta}_{\mathrm{M}}|}\boldsymbol{\Theta}_{\mathrm{M}} + \frac{1 - \cos|\boldsymbol{\theta}_{\mathrm{M}}|}{|\boldsymbol{\theta}_{\mathrm{M}}|^2}\boldsymbol{\Theta}_{\mathrm{M}}^2$$

where $\boldsymbol{\Theta}_{\mathrm{M}}$ is the skew symmetric representation of the midpoint rotation vector.

$$\boldsymbol{\Theta}_{\mathrm{M}} = \begin{bmatrix} 0 & -\theta_{\mathrm{Mz}} & \theta_{\mathrm{My}} \\ \theta_{\mathrm{Mz}} & 0 & -\theta_{\mathrm{M}x} \\ -\theta_{\mathrm{My}} & \theta_{\mathrm{M}x} & 0 \end{bmatrix}$$

The axis directions of the co-rotated coordinate system can now be computed as

$$\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix} = \mathbf{R}_{\mathrm{r}}\begin{bmatrix}\mathbf{r}_1^0 & \mathbf{r}_2^0 & \mathbf{r}_3^0\end{bmatrix}$$

The position of a point on the rigidly rotated axis of the element can be obtained as

$$\mathbf{x}_{\mathrm{R}} = \mathbf{x}_{\mathrm{M}} + \mathbf{R}_{\mathrm{r}}(\mathbf{X}_2 - \mathbf{X}_1)\left(\xi - \frac{1}{2}\right)$$

where the local coordinate ξ ranges from 0 to 1, and $\mathbf{X}_i$ denotes original node coordinates. $\mathbf{x}_{\mathrm{M}}$ is the midpoint position, computed as the average of the two nodes,

$$\mathbf{x}_{\mathrm{M}} = \frac{1}{2}(\mathbf{X}_1 + \mathbf{u}_1 + \mathbf{X}_2 + \mathbf{u}_2)$$

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

$$\bar{\mathbf{u}} = \mathbf{X} + \mathbf{u} - \mathbf{x}_{\mathrm{R}} = \mathbf{u} - \mathbf{u}_{\mathrm{R}}$$

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.

$$\mathbf{u} = \mathbf{u}_\mathrm{R} + \bar{\mathbf{u}}$$

A deformational rotation matrix $\bar{\mathbf{R}}$ can be defined as

$$\bar{\mathbf{R}} = \mathbf{I} + \bar{\Theta}$$

where $\bar{\boldsymbol{\Theta}}$ is the skew-symmetric representation of the deformational rotation vector.

The total rotation vector is computed from a total rotation matrix, **R**. The total rotation matrix is first composed from the rigid body rotation and the incremental rotation.

$$\mathbf{R} = \bar{\mathbf{R}}\mathbf{R}_\mathrm{r}$$

The total rotation vector can now be extracted from the total rotation matrix. The magnitude of the rotation vector is first computed as

$$\theta = \mathrm{acos}\left(\frac{\mathrm{trace}(\mathbf{R}) - 1}{2}\right)$$

The full rotation vector is then computed as

$$\boldsymbol{\theta} = \frac{\theta}{2\sin\theta}\begin{bmatrix} R_{32}-R_{23} \\ R_{13}-R_{31} \\ R_{21}-R_{12} \end{bmatrix}$$

To avoid singularity problems when the angle is close to zero, the gamma function is actually used in the expressions, since

$$\frac{\theta}{\sin\theta} = \gamma\left(1+\frac{\theta}{\pi}\right)\gamma\left(1-\frac{\theta}{\pi}\right)$$

Strain-Displacement/Rotation Relation

The axial strain depends on the rotation derivative (curvature) and axial displacement derivative defined by the shape function and the transversal coordinate in the beam. For the 3D case it becomes

$$\varepsilon = z_l\frac{\partial\theta_{ly}}{\partial s} - y_l\frac{\partial\theta_{lz}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}$$

The coordinates from the beam center line in the local transversal directions are denoted z_l and y_l respectively. In the 2D case, the first term is omitted, and the local z direction is always directed out of the plane.

The total strain ε consists of thermal ($\varepsilon_{\mathrm{th}}$), hygroscopic ($\varepsilon_{\mathrm{hs}}$), initial ($\varepsilon_{\mathrm{i}}$), and elastic strains($\varepsilon_{\mathrm{el}}$)

$$\varepsilon = \varepsilon_{\mathrm{el}} + \varepsilon_{\mathrm{th}} + \varepsilon_{\mathrm{hs}} + \varepsilon_{\mathrm{i}}$$

Stress-Strain Relation

The stress-strain relation for the axial deformation in the beam is described by

$$\sigma = E\varepsilon_{\mathrm{el}} + \boldsymbol{\sigma}_{\mathrm{i}}$$

where $\boldsymbol{E}$ is Young's modulus, and σ_{i} is the initial stress.

The stress strain relation for the torsional and shear deformation is

$$\tau = G\gamma$$

where τ is the shear stress, γ is the shear strain and G is the shear modulus. Often the material data is given in terms of Young's modulus and Poisson's ratio ν, in which case G is defined as

$$G = \frac{E}{2(1+\nu)}$$

Thermal Strain

The temperature is assumed to vary linearly across the beam's cross section. For the 3D beam it becomes

$$T = T_{\mathrm{m}} + T_{\mathrm{gz}} z_1 + T_{\mathrm{gy}} y_1$$

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

$$\varepsilon_{\mathrm{th}} = \alpha(T_{\mathrm{m}} + T_{\mathrm{g}z} z_1 + T_{\mathrm{g}y} y_1 - T_{\mathrm{ref}})$$

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

$$c_{\mathrm{mo}} = c_{\mathrm{m}} + c_{\mathrm{gz}} z_1 + c_{\mathrm{gy}} y_1$$

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

$$\varepsilon_{\mathrm{th}} = \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}} z_1 + c_{\mathrm{gy}} y_1 - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, and $c_{\mathrm{mo,ref}}$ is the strain-free reference moisture concentration.

For the 2D beam, the term depending on z_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.

$$M_{\mathrm{yi}} = \int_A \sigma_{\mathrm{i}} z_1 dA$$

$$M_{\mathrm{zi}} = -\int_A \sigma_{\mathrm{i}} y_1 dA$$

$$N_{\mathrm{i}} = \int_A \sigma_{\mathrm{i}} dA$$

$$M_{\mathrm{xi}} = \int_A (\tau_{\mathrm{i}xz} y_1 - \tau_{\mathrm{i}yz} z_1) dA$$

In 2D the x and y components of moments disappear.

In a Timoshenko beam, the initial shear forces appear as independent quantities, and can also be specified.

$$T_{\mathrm{yi}} = \int_A \tau_{\mathrm{i}xy} dA$$

$$T_{\mathrm{zi}} = \int_A \tau_{\mathrm{i}xz} dA$$

The initial strain is the strain before any loads, displacements, and initial stresses have been applied. The initial axial strain distribution is given as initial curvature and initial axial strain

$$\varepsilon_{\mathrm{i}} = z_1\left(\frac{\partial\theta_{1y}}{\partial s}\right)_{\mathrm{i}} - y_1\left(\frac{\partial\theta_{1z}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}$$

In 2D the z_1 dependent term disappears. As initial strain for the torsional degree of freedom, the derivative of the twist angle,

$$\left(\frac{\partial\theta_{1x}}{\partial s}\right)_{\mathrm{i}}$$

is used.

In a Timoshenko beam it is also possible to set the initial values for the shear strains.

Implementation

The implementation is based on the principle of virtual work, which states that the sum of virtual work from internal strains and external loads equals zero:

$$\delta W = \delta\int_V (-\varepsilon_{\mathrm{el}}\sigma + \mathbf{u}^T\mathbf{F}dV) = 0$$

The beam elements are formulated in terms of the stress resultants (normal force, bending moments and twisting moment).

The normal force is defined as

$$N = \int_A \sigma dA = \int_A (E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}})dA =$$

$$\int_A \left(E\left[\left(z_{\mathrm{l}}\frac{\partial\theta_{\mathrm{l}y}}{\partial s} - y_{\mathrm{l}}\frac{\partial\theta_{\mathrm{l}z}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left(z_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{l}y}}{\partial s}\right)_{\mathrm{i}} - y_{\mathrm{l}}\left(\frac{\partial\theta_{\mathrm{l}z}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) -\right.\right.$$

$$\left.\left.\alpha(T_{\mathrm{m}} + T_{\mathrm{g}z}z_{\mathrm{l}} + T_{\mathrm{g}y}y_{\mathrm{l}} - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}}z_{\mathrm{l}} + c_{\mathrm{gy}}y_{\mathrm{l}} - c_{\mathrm{mo,ref}})\right] + \sigma_{\mathrm{i}}\right)dA =$$

$$\int_A \left(E\left[\left(\frac{\partial u_{\mathrm{axi}}}{\partial s} - \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) - \alpha(T_{\mathrm{m}} - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} - c_{\mathrm{mo,ref}})\right]\right)dA + \int_A \sigma_{\mathrm{i}}dA =$$

$$EA\left[\left(\frac{\partial u_{\mathrm{axi}}}{\partial s} - \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) - \alpha(T_{\mathrm{m}} - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} - c_{\mathrm{mo,ref}})\right] + N_{\mathrm{i}}$$

Because the local coordinates are defined with their origin at the centroid of the cross section, any surface integral of an odd power of a local coordinate evaluates to zero.

The beam bending moments are defined as

$$M_{1y} = \int_A \sigma z_1 dA = \int_A z_1(E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}})dA =$$

$$\int_A z_1\Bigg(E\Bigg[\left(z_1\frac{\partial\theta_{1y}}{\partial s} - y_1\frac{\partial\theta_{1z}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left(z_1\left(\frac{\partial\theta_{1y}}{\partial s}\right)_{\mathrm{i}} - y_1\left(\frac{\partial\theta_{1z}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right) -$$

$$\alpha(T_{\mathrm{m}} + T_{gz}z_1 + T_{\mathrm{gy}}y_1 - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}}z_1 + c_{\mathrm{gy}}y_1 - c_{\mathrm{mo,ref}})\Bigg] + \sigma_{\mathrm{i}}\Bigg)dA =$$

$$\int_A z_1^{\,2}\left(E\left[\frac{\partial\theta_{1y}}{\partial s} - \left(\frac{\partial\theta_{1y}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{gz} - \beta_h c_{\mathrm{gz}}\right]\right)dA + \int_A \sigma_{\mathrm{i}} z_1 dA =$$

$$EI_{yy}\left[\frac{\partial\theta_{1y}}{\partial s} - \left(\frac{\partial\theta_{1y}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{gz} - \beta_h c_{\mathrm{gz}}\right] + M_{\mathrm{i}y}$$

$$M_{1z} = -\int_A \sigma y_1 dA = -\int_A y_1(E\varepsilon_{\mathrm{el}} + \sigma_{\mathrm{i}})dA =$$

$$-\int_A y_1\Bigg(E\Bigg[\left(z_1\frac{\partial\theta_{1y}}{\partial s} - y_1\frac{\partial\theta_{1z}}{\partial s} + \frac{\partial u_{\mathrm{axi}}}{\partial s}\right) - \left(z_1\left(\frac{\partial\theta_{1y}}{\partial s}\right)_{\mathrm{i}} - y_1\left(\frac{\partial\theta_{1z}}{\partial s}\right)_{\mathrm{i}} + \left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right)_{\mathrm{i}}\right)$$

$$-\alpha(T_{\mathrm{m}} + T_{\mathrm{gz}}z_1 + T_{\mathrm{gy}}y_1 - T_{\mathrm{ref}}) - \beta_h(c_{\mathrm{m}} + c_{\mathrm{gz}}z_1 + c_{\mathrm{gy}}y_1 - c_{\mathrm{mo,ref}})\Bigg] + \sigma_{\mathrm{i}}\Bigg)dA =$$

$$-\int_A y_1^{\,2}\left(E\left[-\frac{\partial\theta_{1z}}{\partial s} + \left(\frac{\partial\theta_{1z}}{\partial s}\right)_{\mathrm{i}} - \alpha T_{gy} - \beta_h c_{\mathrm{gy}}\right]\right)dA - \int_A \sigma_{\mathrm{i}} y_1 dA =$$

$$EI_{zz}\left[\frac{\partial\theta_{1z}}{\partial s} - \left(\frac{\partial\theta_{1z}}{\partial s}\right)_{\mathrm{i}} + \alpha T_{gy} + \beta_h c_{\mathrm{gy}}\right] + M_{\mathrm{i}z}$$

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

$$M_{1x} = GJ\left(\frac{\partial\theta_{1x}}{\partial s} - \left(\frac{\partial\theta_{1x}}{\partial s}\right)_{\mathrm{i}}\right) + M_{\mathrm{i}x}$$

Using the beam moment and normal force the expression for the virtual work becomes very compact:

$$\delta W = -\int_L\left(M_{1y}\delta\left(\frac{\partial\theta_{1y}}{\partial s}\right) + M_{1z}\delta\left(\frac{\partial\theta_{1z}}{\partial s}\right) + N\delta\left(\frac{\partial u_{\mathrm{axi}}}{\partial s}\right) + M_{1x}\delta\left(\frac{\partial\theta_{1x}}{\partial s}\right)\right)dx$$

For 2D, the first and fourth terms are omitted. For the case of Timoshenko beam, there is also a shear stress contribution added,

$$\delta W = -\int_{L}(T_{1y}\delta\gamma_{xy} + T_{1z}\delta\gamma_{xz})dx$$

were 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

$$\sigma_{\mathrm{n}} = \frac{N}{A}$$

The normal stress from bending is computed in four user-selected points (y_{1k}, z_{1k}) in the cross section as

$$\sigma_{\mathrm{b}k} = \frac{M_{1y}z_{1k}}{I_{yy}} - \frac{M_{1z}y_{1k}}{I_{zz}}$$

In 2D, only two points, specified by their local y-coordinates are used.

The total normal stress in these points is then

$$\sigma_k = \sigma_{\mathrm{b}k} + \sigma_{\mathrm{n}}$$

The peak normal stress in the section is defined as

$$\sigma_{\max} = \max(|\sigma_k|)$$

When using the built in common cross sections, a special method is used for the Circular and Pipe sections. Since there are no extreme positions around a circle, a maximum bending stress is computed as

$$\sigma_{\mathrm{b,\,max}} = \frac{\mathrm{d_o}\sqrt{M_{1y}^2 + M_{1z}^2}}{2I_{zz}}$$

where $\mathrm{d_o}$ 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

$$\tau_{\mathrm{t,\,max}} = \frac{|M_{1x}|}{W_{\mathrm{t}}}$$

where W_{t} is the torsional section modulus. This result is available only in 3D.

The section shear forces are computed in two different ways depending on the beam formulation. For Euler-Bernoulli theory, the section forces proportional to the third derivative of displacement, or equivalently, the second derivative of the rotation.

$$T_{1z} = EI_{yy}\frac{\partial^2\theta_{1y}}{\partial s^2}$$

$$T_{1y} = -EI_{zz}\frac{\partial^2\theta_{1z}}{\partial s^2}$$

where T_{1z} is available only in 3D. In the case of Timoshenko theory shear force is computed directly from the shear strain.

The average shear stresses are computed from the shear forces as

$$\begin{aligned}\tau_{sz,\,\mathrm{ave}} &= \frac{T_{1z}}{A}\\ \tau_{sy,\,\mathrm{ave}} &= \frac{T_{1y}}{A}\end{aligned} \qquad (7\text{-}1)$$

Since the shear stresses are not constant over the cross section, the maximum shear stresses are also available, using section dependent correction factors:

$$\begin{aligned} \tau_{\mathrm{s}z,\,\max} &= \mu_z \tau_{\mathrm{s}z,\,\mathrm{ave}} \\ \tau_{\mathrm{s}y,\,\max} &= \mu_y \tau_{\mathrm{s}y,\,\mathrm{ave}} \end{aligned} \qquad (7\text{-}2)$$

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

$$\begin{aligned} \tau_{xz,\,\max} &= |\tau_{\mathrm{s}z,\,\max}| + \tau_{\mathrm{t},\,\max} \\ \tau_{xy,\,\max} &= |\tau_{\mathrm{s}y,\,\max}| + \tau_{\mathrm{t},\,\max} \end{aligned}$$

The maximum von Mises effective stress for the cross section is the defined as

$$\sigma_{\mathrm{mises}} = \sqrt{\sigma^2_{\max} + 3\tau^2_{xy,\,\max} + 3\tau^2_{xz,\,\max}}$$

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

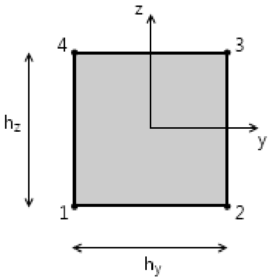


Figure 7-2: Geometry of a rectangular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-2: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
A	$h_{\mathrm{y}}h_{\mathrm{z}}$	
I_{zz}	$\dfrac{h_{\mathrm{y}}^3h_{\mathrm{z}}}{12}$	
e_{z}	0	
μ_{y}	1.5	
κ_{y}	5/6	
I_{yy}	$\dfrac{h_{\mathrm{z}}^3h_{\mathrm{y}}}{12}$	
e_{y}	0	
μ_{z}	1.5	
κ_{z}	5/6	
J	$\dfrac{h_{\mathrm{y}}^2h_z^2q}{3}\left(1-\sum_{n=1}^{2}\dfrac{192q}{\pi^5(2n-1)^5}\tanh\left(\dfrac{(2n-1)\pi}{2q}\right)\right)$	q= min(hy/hz,hz/hy)

TABLE 7-2: RECTANGULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
W_t	$\dfrac{J}{\sqrt{h_\mathrm{y}h_zq}\left(1-\displaystyle\sum_{n=1}^{2}\dfrac{8}{\pi^2(2n-1)^2\cosh\left(\dfrac{(2n-1)\pi}{2q}\right)}\right)}$	q= min(hy/hz,hz/hy)
p_1	$\left(\dfrac{-h_\mathrm{y}}{2},\dfrac{-h_\mathrm{z}}{2}\right)$	
p_2	$\left(\dfrac{h_\mathrm{y}}{2},\dfrac{-h_\mathrm{z}}{2}\right)$	
p_3	$\left(\dfrac{h_\mathrm{y}}{2},\dfrac{h_\mathrm{z}}{2}\right)$	
p_4	$\left(\dfrac{-h_\mathrm{y}}{2},\dfrac{h_\mathrm{z}}{2}\right)$	
r_e	$\dfrac{h_\mathrm{y}+h_\mathrm{z}}{4}$	

BOX SECTION

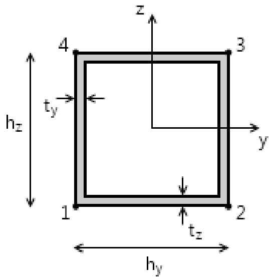


Figure 7-3: Geometry of a box shaped cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-3: BOX SECTION CONSTANTS

PROPERTY	**EXPRESSION**	**REMARKS**
A	$2(h_y t_z + h_z t_y) - 4 t_y t_z$	
I_{zz}	$\dfrac{t_z h_y^3 + t_y^3 (h_z - 2t_z)}{6} + \dfrac{t_y (h_z - 2t_z)(h_y - t_y)^2}{2}$	
e_z	0	
μ_y	$\dfrac{(h_y^2 h_z - (h_y - 2t_y)^2 (h_z - 2t_z))A}{16 t_z I_{zz}}$	
κ_y	$\dfrac{2h_y t_z}{A}$	
I_{yy}	$\dfrac{t_y h_z^3 + t_z^3 (h_y - 2t_y)}{6} + \dfrac{t_z (h_y - 2t_y)(h_z - t_z)^2}{2}$	
e_y	0	
μ_z	$\dfrac{(h_z^2 h_y - (h_z - 2t_z)^2 (h_y - 2t_y))A}{16 t_y I_{yy}}$	
κ_z	$\dfrac{2h_z t_y}{A}$	
J	$\dfrac{2(h_y - t_y)^2 (h_z - t_z)^2}{\dfrac{h_y - t_y}{t_z} + \dfrac{h_z - t_z}{t_y}}$	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	$\left(\dfrac{-h_y}{2}, \dfrac{-h_z}{2}\right)$	
p_2	$\left(\dfrac{h_y}{2}, \dfrac{-h_z}{2}\right)$	
p_3	$\left(\dfrac{h_y}{2}, \dfrac{h_z}{2}\right)$	

TABLE 7-3: BOX SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
p_4	$\left(\frac{-h_y}{2}, \frac{h_z}{2}\right)$	
r_e	$\frac{h_y + h_z}{4}$	

CIRCULAR SECTION

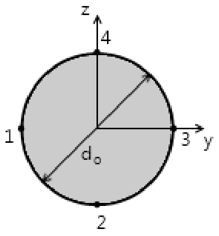


Figure 7-4: Geometry of a circular cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-4: 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	

TABLE 7-4: CIRCULAR SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
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

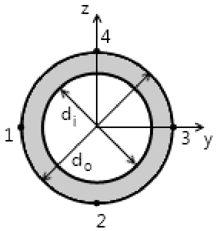


Figure 7-5: Geometry of a pipe cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-5: 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	
J	$\frac{\pi(d_o^4 - d_i^4)}{32}$	
W_t	$\frac{\pi(d_o^4 - d_i^4)}{16d_o}$	
p_1	$\left(\frac{-d_o}{2}, 0\right)$	
p_2	$\left(0, \frac{-d_o}{2}\right)$	
p_3	$\left(\frac{d_o}{2}, 0\right)$	

TABLE 7-5: PIPE SECTION CONSTANTS

PROPERTY	FORMULA	REMARKS
p_4	$\left(0, \frac{d_\mathrm{o}}{2}\right)$	
r_e	$\frac{d_\mathrm{o}}{2}$	

H-PROFILE SECTION

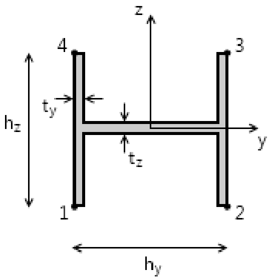


Figure 7-6: Geometry of a an H-profile cross section. The diagram also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-6: H-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
A	$2h_\mathrm{z}t_\mathrm{y} + t_\mathrm{z}(h_\mathrm{y} - 2t_\mathrm{y})$	
I_zz	$\frac{2h_\mathrm{z}t_\mathrm{y}^3 + t_\mathrm{z}(h_\mathrm{y} - 2t_\mathrm{y})^3}{12} + \frac{t_\mathrm{y}h_\mathrm{z}(h_\mathrm{y} - t_\mathrm{y})^2}{2}$	
e_z	0	
μ_y	$\frac{[4h_\mathrm{z}t_\mathrm{y}(h_\mathrm{y} - t_\mathrm{y}) + t_\mathrm{z}(h_\mathrm{y} - 2t_\mathrm{y})^2]A}{8t_\mathrm{z}I_\mathrm{zz}}$	
κ_y	$\frac{h_\mathrm{y}t_\mathrm{z}}{A}$	

TABLE 7-6: H-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
I_{yy}	$\frac{2t_{\mathrm{y}}h_{\mathrm{z}}^3 + t_{\mathrm{z}}^3(h_{\mathrm{y}} - 2t_{\mathrm{y}})}{12}$	
e_{y}	0	
μ_{z}	$\frac{(h_{\mathrm{z}}^2 - t_{\mathrm{z}}^2)A}{8I_{\mathrm{yy}}}$	
κ_{z}	$\frac{8h_z t_y}{3A}$	
J	$\frac{2t_{\mathrm{y}}^3h_{\mathrm{z}} + t_{\mathrm{z}}^3(h_{\mathrm{y}} - 2t_{\mathrm{y}})}{3}$	Thin-walled approximation
W_{t}	$\frac{J}{\max(t_{\mathrm{y}}, t_{\mathrm{z}})}$	Thin-walled approximation
p_1	$\left(\frac{-h_{\mathrm{y}}}{2}, \frac{-h_{\mathrm{z}}}{2}\right)$	
p_2	$\left(\frac{h_{\mathrm{y}}}{2}, \frac{-h_{\mathrm{z}}}{2}\right)$	
p_3	$\left(\frac{h_{\mathrm{y}}}{2}, \frac{h_{\mathrm{z}}}{2}\right)$	
p_4	$\left(\frac{-h_{\mathrm{y}}}{2}, \frac{h_{\mathrm{z}}}{2}\right)$	
r_{e}	$\frac{h_{\mathrm{y}} + h_{\mathrm{z}}}{4}$	

U-PROFILE SECTION

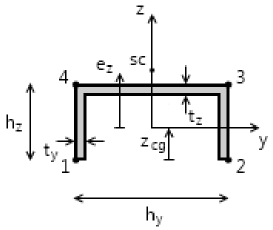


Figure 7-7: Geometry of a U-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-7: 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	$\frac{(h_y^2 h_z – (h_y - 2t_y)^2(h_z - t_z))A}{8t_z I_{zz}}$	
κ_y	$\frac{h_y t_z}{A}$	
I_{yy}	$\frac{8t_y h_z^3 + t_z^3(h_y - 2t_y) + 3t_z(h_y - 2t_y)(2h_z - t_z)^2}{12}$ $- z_{CG}{}^2 A$	
e_y	0	

TABLE 7-7: U-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
μ_z	$\dfrac{z_\mathrm{CG}{}^2A}{2I_\mathrm{yy}}$	
κ_z	$\dfrac{8h_zt_y}{3A}$	
J	$\dfrac{2t_\mathrm{y}^3h_\mathrm{z}+t_\mathrm{z}^3(h_\mathrm{y}-2t_\mathrm{y})}{3}$	Thin-walled approximation
W_t	$\dfrac{J}{\max(t_\mathrm{y},t_\mathrm{z})}$	Thin-walled approximation
p_1	$\left(\dfrac{-h_\mathrm{y}}{2},-z_\mathrm{CG}\right)$	
p_2	$\left(\dfrac{h_\mathrm{y}}{2},-z_\mathrm{CG}\right)$	
p_3	$\left(\dfrac{h_\mathrm{y}}{2},h_\mathrm{z}-z_\mathrm{CG}\right)$	
p_4	$\left(\dfrac{-h_\mathrm{y}}{2},h_\mathrm{z}-z_\mathrm{CG}\right)$	
r_e	$\dfrac{h_\mathrm{y}+h_\mathrm{z}}{4}$	

T-PROFILE SECTION

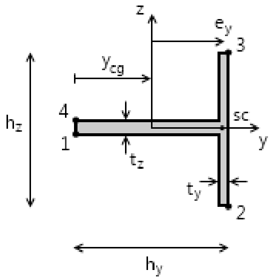


Figure 7-8: Geometry of a T-profile cross section. Also displays in COMSOL Multiphysics when this option is selected.

TABLE 7-8: 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 7-8: T-PROFILE SECTION CONSTANTS

PROPERTY	EXPRESSION	REMARKS
μ_{z}	$\frac{(h_{\mathrm{z}}^2 - t_{\mathrm{z}}^2)A}{8I_{\mathrm{yy}}}$	
κ_{z}	$\frac{4h_z t_y}{3A}$	
J	$\frac{t_{\mathrm{y}}^3 h_{\mathrm{z}} + t_{\mathrm{z}}^3(h_{\mathrm{y}} - t_{\mathrm{y}})}{3}$	Thin-walled approximation
W_{t}	$\frac{J}{\max(t_{\mathrm{y}}, t_{\mathrm{z}})}$	Thin-walled approximation
p_1	$\left(-y_{\mathrm{CG}}, \frac{-t_{\mathrm{z}}}{2}\right)$	
p_2	$\left(h_{\mathrm{y}} - y_{\mathrm{CG}}, \frac{-h_{\mathrm{z}}}{2}\right)$	
p_3	$\left(h_{\mathrm{y}} - y_{\mathrm{CG}}, \frac{h_{\mathrm{z}}}{2}\right)$	
p_3	$\left(-y_{\mathrm{CG}}, \frac{t_{\mathrm{z}}}{2}\right)$	
r_{e}	$\frac{h_{\mathrm{y}} + h_{\mathrm{z}}}{4}$	

The Beam Interface

The **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** $\mathbf{x}_{\mathrm{ref}}$. 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 **u**, which has two components (`u`, `v`) in 2D and three components (`u`, `v`, and `w`) in 3D.
- The rotation angle θ, which has one component in 2D (`th`) and three components in 3D (`thx`, `thy`, and `thz`).

The names can be changed but the names of fields and dependent variables must be unique within a model.

The dependent variable names remain same in both a geometrically linear and a geometrically nonlinear analysis. Under geometric nonlinearity, the dependent variables are however not defined though shape functions. The equivalent shape function variables are (`beam.uLinx`, `beam.uLiny`, `beam.uLinz`) and (`beam.thLinx`, `beam.thLiny`, `beam.thLinz`). In this case, you will see the latter names under Dependent variables in the Solver 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
- Gravity[1]
- Predeformation
- Rigid Domain
- Rotating Frame[1]
- 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**, or **T-profile**. 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

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

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 7-1 and Equation 7-2.

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	1
3D only	Section height in local z direction	h_z	m
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	1

From Specified Points

For **From specified points** enter values in the **Evaluation points in local system** table as needed for the space dimension. Then enter the following parameters in the applicable fields.

COMMENT	DESCRIPTION	PARAMETER	SI UNIT
2D and 3D	Max shear stress factor in local y direction	μ_y	1
3D only	Torsional section modulus	W_t	m^3
3D only	Max shear stress factor in local z direction	μ_z	1

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 ($\mathbf{e}_{\mathrm{xl}}$, $\mathbf{e}_{\mathrm{yl}}$, $\mathbf{e}_{\mathrm{zl}}$) is formed using the following algorithm:

$$\mathbf{v}_{\mathrm{zl}} = \mathbf{e}_{\mathrm{xl}} \times (\mathbf{p} - \mathbf{m})$$

$$\mathbf{e}_{\mathrm{zl}} = \frac{\mathbf{v}_{\mathrm{zl}}}{|\mathbf{v}_{\mathrm{zl}}|}$$

$$\mathbf{e}_{\mathrm{yl}} = \mathbf{e}_{\mathrm{zl}} \times \mathbf{e}_{\mathrm{xl}}$$

Here, **p** is the reference point, and **m** is the midpoint of the beam element. The definition of the local coordinate system is illustrated in Figure 7-9.

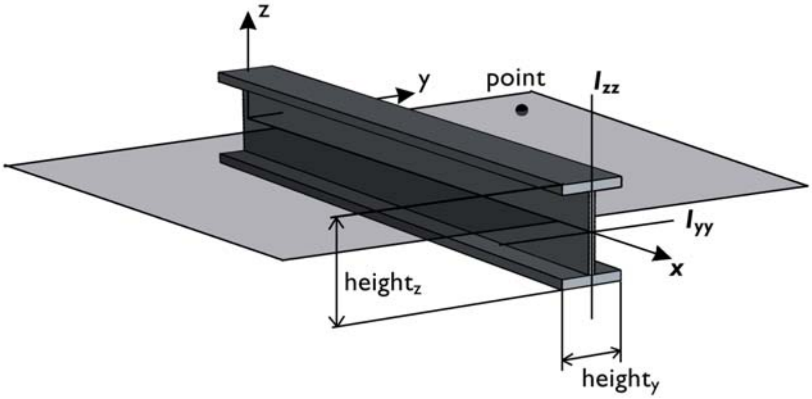


Figure 7-9: Local beam coordinate system defined by a reference point

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 ($\mathbf{e}_{xl}$, $\mathbf{e}_{yl}$, $\mathbf{e}_{zl}$) is formed using the following algorithm:

$$\mathbf{v}_{\mathrm{z1}} = \mathbf{e}_{\mathrm{x1}} \times \mathbf{V}$$

$$\mathbf{e}_{\mathrm{z1}} = \frac{\mathbf{v}_{\mathrm{z1}}}{|\mathbf{v}_{\mathrm{z1}}|}$$

$$\mathbf{e}_{\mathrm{y1}} = \mathbf{e}_{\mathrm{z1}} \times \mathbf{e}_{\mathrm{x1}}$$

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

$$\mathbf{e}_{\mathrm{y1}} = \mathbf{e}'_{\mathrm{y1}}\cos\phi - \mathbf{e}'_{\mathrm{z1}}\sin\phi$$

$$\mathbf{e}_{\mathrm{z1}} = \mathbf{e}'_{\mathrm{z1}}\cos\phi + \mathbf{e}'_{\mathrm{y1}}\sin\phi$$

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

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

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

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.

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.

Enter a value or expression of the **Strain reference temperature** T_{ref}, which is the reference temperature where the thermal strain is zero.

THERMAL BENDING

Enter the **Temperature gradient in local y-direction** T_{gy} (in 2D and 3D) and in the **Temperature gradient in local z-direction** T_{gz} (in 3D), which affects the thermal bending. If beam cross section dimensions have been defined at **Bending stress evaluation points—From section heights**, these could be used in an expression containing the temperature difference.

LOCATION IN USER INTERFACE

Context Menus

Beam>Linear Elastic Material>Thermal Expansion

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree: **Attributes>Thermal Expansion**

Hygroscopic Swelling

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration. It is possible to model bending due to a concentration gradient in the transverse directions of the beam. The concentration is then assumed to vary linearly through the thickness.

MODEL INPUTS

From the **Concentration** c list, select an existing concentration variable from another physics interface, if any concentration variables exist. For **User defined** enter a value or expression for the concentration. This is the centerline concentration of the beam, controlling the axial part of the hygroscopic swelling.

The unit for the input depends on the setting of **Concentration type** in the **Hygroscopic Swelling Properties** section. Only concentration variables having the chosen physical dimension are available in the **Concentration** list.

HYGROSCOPIC SWELLING PROPERTIES

In the **Concentration type** list, select **Molar concentration** (the default) or **Mass concentration**, depending on the units used for the concentration.

Enter a **Strain reference concentration** c_{ref}. This is the concentration at which there are no strains due to hygroscopic swelling.

If **Molar concentration** is selected as **Concentration type**, also enter the **Molar mass** of the fluid, M_{m}. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h}.The default value for the **User defined** case is `1.5e-4` m^3/kg.

HYGROSCOPIC BENDING

Enter the **Concentration gradient in local y-direction** c_{gy} (in 2D and 3D) and in the **Concentration gradient in local z-direction** c_{gz} (in 3D), which affects the hygroscopic bending. If beam cross section dimensions have been defined at **Bending stress evaluation points—From section heights**, these could be used in an expression containing the concentration difference.

LOCATION IN USER INTERFACE

Context Menus

Beam>Linear Elastic Material>Hygroscopic Swelling

Ribbon

Physics tab with **Linear Elastic Material** node selected in the model tree:

Attributes>Hygroscopic Swelling

Initial Stress and Strain

You can add the **Initial Stress and Strain** to 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** $M_{\mathrm{i}z}$ and for 3D models, $M_{\mathrm{i}y}$.
- *For 3D models*: **Initial torsional moment** $M_{\mathrm{i}x}$.
- *For Timoshenko beam*: **Initial shear force** $T_{\mathrm{i}y}$ and for 3D models, $T_{\mathrm{i}z}$.

INITIAL STRAIN

Specify the initial strain as the:

- **Initial axial strain** e_ni.

- **Initial curvature** 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, to define a prescribed rotation 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 θ_{0x}, θ_{0y}, or θ_{0z} field.

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.

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 Acceleratio**n 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.

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

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

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 **Strain 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.

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	$\mathbf{F}_L$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Force per unit volume	$\mathbf{F}_V$	N/m^3	edges boundaries	3D (x, y, z) 2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	edges boundaries	3D (x, y, z) 2D (x, y)

- When **Force per 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 $\mathbf{F}_\mathrm{P}$.

POINT MOMENT

Enter values or expressions for the components (x, y, z for 3D models and z for 2D models) of the point moment $\mathbf{M}_\mathrm{P}$ (3D) or $M_{\mathrm{l}z}$ (2D).

- The $\mathbf{F}_\mathrm{P}$ 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

Solid Connection

> !
>
> The **Solid Connection** node is obsolete, and has been superseded by the Solid-Beam Connection multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.
>
> You are advised to update your model to use the new multiphysics coupling instead, since the **Solid Connection** node will be removed in future versions.

Use a **Solid Connection** node for connecting beams to solids in 2D models. To complete the connection, a corresponding **Beam Connection** node must also be added in the Solid Mechanics interface. The **Solid Connection** node can be added to either points or boundaries.

When using it for a point, the connection is intended for a transition from a beam to a solid, so that beam theory is assumed to be valid on both sides of the connection.

When using it for a boundary, the beam acts as a cladding or stiffener on the solid.

SOLID CONNECTION (POINTS ONLY)

Select a **Connection type**—**Softened** or **Simplified**. When using **Softened** (the default), three extra degrees of freedom are added to each selected point. This allows for a more accurate description of the transition, but the model can in some cases become underconstrained if the mesh on the solid is very coarse. The **Simplified** version of the connection just adds constraints to the boundary of the solid, and will in general cause local disturbances of the stress field.

When using the **Solid Connection** node for a boundary, the connection is intended for placing a beam as a stiffener or cladding along the boundary of a solid. No other settings than the boundary selection are required.

- For more information about coupling different element types, see Coupling Techniques.
- For details about the formulation of this coupling, see Connection Between Shells and Solids

LOCATION IN USER INTERFACE

Context Menus

Beam>Connections>Solid Connection

Ribbon

Physics tab with **Shell** selected:

Boundaries>Connections>Solid Connection

Points>Connections>Solid Connection

Shell Connection

The **Shell Connection** node is obsolete, and has been superseded by the Shell-Beam Connection multiphysics coupling. It cannot be added in version 5.3 and later, but may be present in models created by earlier versions of the software.

You are advised to update your model to use the new multiphysics coupling instead, since the **Shell Connection** node will be removed in future versions.

A beam can be connected to a shell by adding a **Shell Connection** node in the Beam interface and a Beam Connection node in the Shell interface. The connection can be between:

- two edges, one in each physics interface,
- a point in the Beam interface and a boundary in the Shell interface, or
- a point in the Beam interface and an edge in the Shell interface.

There are no settings for the **Shell Connection** node except the selections of points or edges.

- For more information about coupling different element types, see Coupling Techniques.
- For details about the formulation of this coupling, see Connection Between Shells and Beams

LOCATION IN USER INTERFACE

Context Menus

Beam>Connections>Shell Connection

Ribbon

Physics tab with **Shell** selected:

Edges>Connections>Shell Connection

Points>Connections>Shell Connection

Rigid Connector

The **Rigid Connector** is a boundary condition for modeling rigid regions and kinematic constraints such as prescribed rigid rotations. The selected points will move as a single rigid object.

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.

$$\mathbf{X}_{\mathrm{C}} = \mathbf{X}_{\mathrm{C,input}} + \mathbf{X}_{\mathrm{offset}}$$

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 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 Attachments is included 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:

$$F_{\text{freq}} = F(f) \cdot \cos(2\pi f t + \phi)$$

PHASE

Add the phase load $\mathbf{F}_{\text{ph}}$ for harmonic loads. Enter the phase for each component of the load in the corresponding fields.

MOMENT LOAD PHASE

Add the phase for the moment load $\mathbf{M}_{\text{ph}}$ for harmonic loads. Enter the phase for each component of the moment load in the corresponding fields.

LOCATION IN USER INTERFACE

Context Menus

Beam>Edge Load>Phase

Beam>Point Load>Phase

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 non-zero 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

8

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

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 Theory for 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 non-standard cross sections with non-trivial 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 8-1). 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_Models/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 8-1:

TABLE 8-1: 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.CGx`	Center of gravity, y coordinate	As above	

TABLE 8-1: 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 (Izz/Iyy)	Moments of Inertia
bcs.I2	Smallest principal moment of inertia	As above	
bcs.Ixx	Moment of inertia around x-axis	-	
bcs.Iyy	Moment of inertia around y-axis	-	
bcs.Ixy	Deviatoric moment of inertia in xy system	-	
bcs.rg	Radius of gyration	-	
bcs.alpha	Angle from x-axis to first principal axis	Rotation of vector around beam axis (ϕ)	Directions of Principal Axes
bcs.mu1	Max shear stress factor, x1 direction	Max shear stress factor in local y/z direction (μy/μz)	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 (Wt)	

TABLE 8-1: 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`	Non-uniform 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 8-2.

After changing data in this section, you do not need to compute the study for this physics interface again once is has been solved. It is sufficient to do an **Update Solution** to get the stress plots updated.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

TABLE 8-2: 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 8-2: 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

Theory for the Beam Cross Section Interface

In this section the theory for the Beam Cross Section interface is described:

- Cross Section Properties
- Computation of Stresses

Cross Section Properties

The following cross section properties computed by the Beam Cross Section interface are described in this section:

- Area
- Center of Gravity
- Moments of Inertia
- Directions of Principal Axes
- Local Coordinates
- Bending Shear Stresses
- Torsion
- Warping

Area

The area is computed as:

$$A = \int_A dA$$

Center of Gravity

The center of gravity is computed as:

$$x_{\mathrm{CG}} = \frac{1}{A}\int_A x\,dA$$

$$y_{\mathrm{CG}} = \frac{1}{A}\int_A y\,dA$$

Moments of Inertia

The moments of inertia in the XY coordinate system are:

$$I_{\mathrm{xx}} = \int_A (y - y_{\mathrm{CG}})^2 dA$$

$$I_{\mathrm{yy}} = \int_A (x - x_{\mathrm{CG}})^2 dA$$

$$I_{\mathrm{xy}} = \int_A (x - x_{\mathrm{CG}})(y - y_{\mathrm{CG}}) dA$$

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:

$$R = \sqrt{\left(\frac{I_{\mathrm{xx}} - I_{\mathrm{yy}}}{R}\right)^2 + I_{\mathrm{xy}}^2}$$

the principal moments of inertia can be expressed as:

$$I_1 = \frac{I_{\mathrm{xx}} + I_{\mathrm{yy}}}{2} + R$$

$$I_2 = \frac{I_{\mathrm{xx}} + I_{\mathrm{yy}}}{2} - R$$

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:

$$\alpha = \frac{1}{2}\mathrm{atan}\left(\frac{-2I_{\mathrm{xy}}}{I_{\mathrm{xx}} - I_{\mathrm{yy}}}\right)$$

When implemented using the `atan2` function, the angle can be correctly evaluated for all rotations, and returns in the interval $-\pi < \alpha < \pi$.

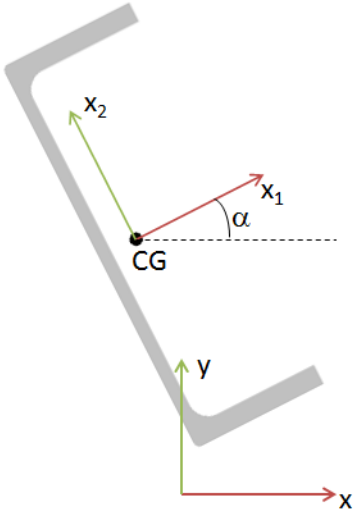


Figure 8-1: Local coordinate system and rotation angle.

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

$$\bar{\gamma} = \frac{\bar{\tau}}{\kappa G} = \frac{T}{\kappa G A} = \frac{T}{G A_s}$$

To compute the shear correction factor, the true strain energy based on the actual stress and strain distribution is set equal to the strain energy from the average shear stress, when acting over the shear area. The full energy expression is

$$\frac{1}{2}\int_A (\tau_{1z}^{(1)}\gamma_{1z}^{(1)} + \tau_{2z}^{(1)}\gamma_{2z}^{(1)})\, dA$$

The strain energy based on the averaged shear stress and shear strain is

$$\frac{1}{2}\overline{\tau_{1z}^{(1)}}\,\overline{\gamma_{1z}^{(1)}}A = \frac{1}{2}\frac{T_1}{A}\overline{\gamma_{1z}^{(1)}}A = \frac{1}{2\kappa_1 G}\left(\frac{T_1}{A}\right)^2 A$$

giving

$$\frac{1}{2G}\int_A ((\tau_{1z}^{(1)})^2 + (\tau_{2z}^{(1)})^2)\,dA = \frac{T_1^2}{2G\kappa_1 A}$$

Since T_1 is a unit shear force, the shear correction factor can be computed as

$$\kappa_1 = \left(A\int_A (\tau^{(1)})^2 dA\right)^{-1}$$

Similarly:

$$\kappa_2 = \left(A\int_A (\tau^{(2)})^2 dA\right)^{-1}$$

Shear Center Distance

The *shear center* (or, equivalently, the *center of rotation*) is the point around which the shear stresses from bending has no torque. In COMSOL it is represented as the distance from the center of gravity of the cross section in the principal axes coordinate system. The torque can be computed as:

$$0 = \int_A (-\tau_{1z}(e_2 - x_2) + \tau_{2z}(e_1 - x_1))\,dA =$$

$$\int_A (\tau_{1z}x_2 + -\tau_{2z}x_1)\,dA + e_1\int_A \tau_{2z}\,dA - e_2\int_A \tau_{1z}\,dA$$

Since there are two separate solutions for the shear stresses, it is possible to split the determination of the two shear center coordinates into:

$$e_1 = -\int_A (\tau_{1z}^{(2)}x_2 + -\tau_{2z}^{(2)}x_1)\,dA$$

$$e_2 = \int_A (\tau_{1z}^{(1)}x_2 + -\tau_{2z}^{(1)}x_1)\,dA$$

Here the fact that the shear force resultant has a unit value has been used.

Derivation of the Equations for Computing the Shear Stresses

Basic beam theory assumptions gives the following stress components:

$$\begin{aligned} \sigma_1 &= 0 \\ \sigma_2 &= 0 \\ \sigma_z &= \frac{M_1 x_2}{I_1} - \frac{M_2 x_1}{I_2} \\ \tau_{12} &= 0 \end{aligned} \tag{8-1}$$

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:

$$\Delta\psi = -\left(\frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2}\right) \tag{8-2}$$

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:

$$\Delta\sigma_{ij} + \frac{1}{1+\nu}\frac{\partial(\mathtt{trace}(\sigma))}{\partial x_i \partial x_j} = 0$$

Given the stress state from Equation 8-1, the only two non-trivial equations are:

$$\Delta\tau_{1z} + \frac{1}{1+\nu}\frac{T_1}{I_2} = 0$$

$$\Delta\tau_{2z} + \frac{1}{1+\nu}\frac{T_2}{I_1} = 0$$

Inserting the assumed stress components gives:

$$\Delta\psi_{,1} + \frac{\nu T_1}{(1+\nu)I_2} + \frac{1}{1+\nu}\frac{T_1}{I_2} = \Delta\psi_{,1} + \frac{T_1}{I_2} = 0$$

$$\Delta\psi_{,2} + \frac{\nu T_2}{(1+\nu)I_1} + \frac{1}{1+\nu}\frac{T_2}{I_1} = \Delta\psi_{,2} + \frac{T_2}{I_1} = 0$$

Integration of the first equation with respect to x_1 and the second equation with respect to x_2 gives:

$$\Delta\psi + \frac{T_1 x_1}{I_2} + f_2(x_2) = 0$$

$$\Delta\psi + \frac{T_2 x_2}{I_1} + f_1(x_1) = 0$$

Combining these two equations results in:

$$\Delta\psi = -\left(\frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2}\right)$$

This is the same equation as Equation 8-2. 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.

$$\int_A \psi_{,1} dA = \int_A \left(\psi_{,1} + x_1\left(\Delta\psi + \frac{T_2 x_2}{I_1} + \frac{T_1 x_1}{I_2}\right)\right) dA =$$

$$\int_A (\psi_{,1} + x_1 \Delta\psi)\, dA + T_1 = \int_A \left(\frac{\partial}{\partial x_1}(x_1 \psi_{,1}) + \frac{\partial}{\partial x_2}(x_1 \psi_{,2})\right)\, dA + T_1 =$$

$$\oint_\Gamma ((x_1 \psi_{,1}) n_1 + (x_1 \psi_{,2}) n_2)\, ds + T_1 = \oint_\Gamma x_1 (\nabla\psi \cdot \mathbf{n}) ds + T_1 =$$

$$\oint_\Gamma x_1 \left(-\left(\frac{\nu T_1 x_2^2}{2(1+\nu) I_2} n_1 + \frac{\nu T_2 x_1^2}{2(1+\nu) I_1} n_2\right)\right) ds + T_1 =$$

$$\int_A \left(-\frac{\nu T_1 x_2^2}{2(1+\nu) I_2}\right) dA + T_1 = \frac{-\nu T_1 I_1}{2(1+\nu) I_2} + T_1$$

In the transformations above these facts are used:

- The area integral of x_1, x_1 or $x_1 x_2$ are zero since the coordinate system is positioned at the center of gravity and is oriented along the principal axes.
- The area integral of x_2^2 is I_1.
- The divergence theorem is used for transformation between area and surface integrals.
- The Neumann condition on the boundary is inserted into the integral along the boundary.

This proves that the assumed stress field also produces the correct resultants.

When internal holes are present, it is necessary ensure compatibility in the sense that the displacement is single valued when going around the hole:

$$\oint_\Gamma du_1 = \oint_\Gamma du_2 = \oint_\Gamma du_z = 0$$

The displacements can be derived from the strains, which are given by the stress state:

$$\varepsilon_1 = \frac{\partial u_1}{\partial x_1} = -\nu\varepsilon_z$$

$$\varepsilon_2 = \frac{\partial u_2}{\partial x_2} = -\nu\varepsilon_z$$

$$\varepsilon_z = \frac{\partial u_z}{\partial z} = \frac{M_1 x_2}{EI_1} - \frac{M_2 x_1}{EI_2}$$

$$\gamma_{12} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} = 0$$

$$\gamma_{1z} = \frac{\partial u_1}{\partial z} + \frac{\partial u_z}{\partial x_1} = \frac{1}{G}\left(\psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2}\right)$$

$$\gamma_{2z} = \frac{\partial u_2}{\partial z} + \frac{\partial u_z}{\partial x_2} = \frac{1}{G}\left(\psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1}\right)$$

Integration of the direct strains gives:

$$u_1 = -\frac{\nu M_1 x_1 x_2}{EI_1} + \frac{\nu M_2 x_1^2}{2EI_2} + g_1(x_2, z)$$

$$u_2 = -\frac{\nu M_1 x_2^2}{2EI_1} + \frac{\nu M_2 x_1 x_2}{EI_2} + g_2(x_1, z)$$

Since the only part of the displacement that is relevant for the bending shear stresses is independent of the z coordinate, the functions g_1 and g_2 can be considered as independent of z.

$$\oint_\Gamma du_1 = \oint_\Gamma\left(\frac{\partial u_1}{\partial x_1}dx_1 + \frac{\partial u_1}{\partial x_2}dx_2\right) =$$

$$\oint_\Gamma\left(\left(-\left(\frac{\nu M_1 x_2}{EI_1} - \frac{\nu M_2 x_1}{EI_2}\right)\right)dx_1 + \left(-\frac{\nu M_1 x_1}{EI_1} + \frac{\partial g_1}{\partial x_2}\right)dx_2\right) =$$

$$\int_A\left(\frac{\nu M_1 x_1 x_2}{EI_1}\right)dA - \int_A\left(\frac{\nu M_1 x_1 x_2}{EI_1}\right)dA = 0$$

In the last transformation Green's theorem is used. The uniqueness of the u_2 displacement can be shown in the same way.

The uniqueness of the out-of-plane component of the displacement is shown in Equation 8-3:

$$\begin{aligned}
&\oint_{\Gamma} du_z = \oint_{\Gamma}\left(\frac{\partial u_z}{\partial x_1}dx_1 + \frac{\partial u_z}{\partial x_2}dx_2\right) = \oint_{\Gamma}\left(\left(\gamma_{1z} - \frac{\partial u_1}{\partial z}\right)dx_1 + \left(\gamma_{2z} - \frac{\partial u_2}{\partial z}\right)dx_2\right) = \\
&\oint_{\Gamma}\left(\frac{1}{G}\left(\psi_{,1} + \frac{\nu T_1 x_2^2}{2(1+\nu)I_2}\right) - \left(\frac{\nu T_1 x_1 x_2}{EI_1} + \frac{\nu T_2 x_1^2}{2EI_2}\right)\right)dx_1 + \\
&\oint_{\Gamma}\left(\frac{1}{G}\left(\psi_{,2} + \frac{\nu T_2 x_1^2}{2(1+\nu)I_1}\right) - \left(\frac{\nu T_1 x_2^2}{2EI_1} + \frac{\nu T_2 x_1 x_2}{EI_2}\right)\right)dx_2 = \\
&-\int_{A}\left(\frac{1}{G}\left(\psi_{,12} + \frac{\nu T_1 x_2}{(1+\nu)I_2}\right) - \frac{\nu T_1 x_1}{EI_1}\right)dA + \\
&\int_{A}\left(\frac{1}{G}\left(\psi_{,21} + \frac{\nu T_2 x_1}{(1+\nu)I_1}\right) - \frac{\nu T_2 x_2}{EI_2}\right)dA = 0
\end{aligned}\tag{8-3}$$

In the last step of Equation 8-3 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:

$$\Delta\psi^{(2)} = -\frac{x_2}{I_1}$$

$$\nabla\psi^{(2)} \cdot \mathbf{n} = \frac{-\nu x_1^2}{2(1+\nu)I_1} n_1$$

$$\tau_{1z}^{(2)} = \psi_{,1}^{(2)}$$
$$\tau_{2z}^{(2)} = \psi_{,2}^{(2)} + \frac{\nu x_1^2}{2(1+\nu)I_1}$$

TORSION

The torsional properties cannot in general be computed using a closed form expression. Determining the torsional rigidity requires the solution of a PDE over the cross section. There are two ways to do this: Using a warping function or using the Prandtl stress function. The Prandtl stress function approach is used in COMSOL 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:

$$\tau_{\mathrm{xz}} = \frac{\partial\phi}{\partial y}$$
$$\tau_{\mathrm{yz}} = -\frac{\partial\phi}{\partial x}$$

The torsional modulus can be determined as:

$$W_{\mathrm{t}} = \frac{J}{max(|\nabla\phi|)}$$

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:

$$\begin{aligned}0 &= \oint_{\Gamma_i} dw = \oint_{\Gamma_i}\left[\frac{\partial w}{\partial x}dx + \frac{\partial w}{\partial y}dy\right] = \oint_{\Gamma_i}\left[\left(\gamma_{\mathrm{xz}} - \left(\frac{\partial u}{\partial z}\right)\right)dx + \left(\gamma_{\mathrm{yz}} - \left(\frac{\partial v}{\partial z}\right)\right)dy\right] \\ &= \oint_{\Gamma_i}\left[\left(\frac{\partial\phi}{\partial y} + y\right)dx + \left(-\frac{\partial\phi}{\partial x} - x\right)dy\right] = \oint_{\Gamma_i}\left[\left(\frac{\partial\phi}{\partial y} + y\right)t_{\mathrm{x}} - \left(\frac{\partial\phi}{\partial x} + x\right)t_{\mathrm{y}}\right]ds\end{aligned}$$

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:

$$\oint_{\Gamma_i}\left[\left(\frac{\partial\phi}{\partial y} + y\right)t_{\mathrm{x}} - \left(\frac{\partial\phi}{\partial x} + x\right)t_{\mathrm{y}}\right]ds = 0$$

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:

$$\nabla\omega \cdot \mathbf{n} = (x - e_\mathrm{x})t_\mathrm{x} + (y - e_\mathrm{y})t_\mathrm{y}$$

The offset by the shear center coordinates $(e_\mathrm{x}, e_\mathrm{y})$ is introduced since the torsion theory assumes that the coordinate system has it 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 constant, which is used in analysis of non-uniform torsion, is defined as:

$$C_\mathrm{w} = \int_A \omega(x,y)^2 dA$$

The axial stress caused by non-uniform torsion is:

$$\sigma_{\mathrm{w}}(x,y) = \frac{B\,\omega(x,y)}{C_{\mathrm{w}}}$$

where B is the *bimoment*. The maximum axial stress is:

$$\sigma_{\mathrm{w,\,max}} = \frac{|B|\max(|\omega(x,y)|)}{C_{\mathrm{w}}} = \frac{|B|}{K_{\mathrm{w}}}$$

The warping modulus is then defined as:

$$K_{\mathrm{w}} = \frac{\max(|\omega(x,y)|)}{C_{\mathrm{w}}}$$

Given the warping constant, it is possible to compute a non-dimensional number that can be used to characterize the influence of non-uniform torsion in a beam with a certain length L. This number is:

$$\frac{\pi^2 EC_{\mathrm{w}}}{GJL^2} = \frac{2(1+\nu)\pi^2 C_{\mathrm{w}}}{JL^2}$$

Since the length is independent of the cross section, the sensitivity number is defined as:

$$s_{\mathrm{w}} = \frac{2(1+\nu)\pi^2 C_{\mathrm{w}}}{J}$$

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:

$$\sigma_{\mathrm{M1}} = \frac{M_1 x_2}{I_1}$$
$$\sigma_{\mathrm{M2}} = -\frac{M_2 x_1}{I_2}$$

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 x1 axis are:

$$\tau_{\mathrm{T1x}} = (\tau_{\mathrm{1z}}^{(1)} \cos\alpha - \tau_{\mathrm{2z}}^{(1)} \sin\alpha) \cdot T_1$$
$$\tau_{\mathrm{T1y}} = (\tau_{\mathrm{2z}}^{(1)} \cos\alpha + \tau_{\mathrm{1z}}^{(1)} \sin\alpha) \cdot T_1$$

The components of the shear stresses caused by a shear force T_2 along the x2 axis are:

$$\tau_{\mathrm{T2x}} = (\tau_{\mathrm{1z}}^{(2)} \cos\alpha - \tau_{\mathrm{2z}}^{(2)} \sin\alpha) \cdot T_2$$
$$\tau_{\mathrm{T2y}} = (\tau_{\mathrm{2z}}^{(2)} \cos\alpha + \tau_{\mathrm{1z}}^{(2)} \sin\alpha) \cdot T_2$$

In both cases, the resultants are also computed as:

$$\tau_{\mathrm{T1}} = \sqrt{\tau_{\mathrm{T1x}}^2 + \tau_{\mathrm{T1y}}^2} \text{ and } \tau_{\mathrm{T2}} = \sqrt{\tau_{\mathrm{T2x}}^2 + \tau_{\mathrm{T2y}}^2}$$

TORSIONAL SHEAR STRESSES

The components of the shear stress caused by Saint-Venant torsion are:

$$\tau_{\mathrm{Mtx}} = \frac{M_{\mathrm{t}}\phi_{,\mathrm{y}}}{J}$$

$$\tau_{\mathrm{Mty}} = -\frac{M_{\mathrm{t}}\phi_{,\mathrm{x}}}{J}$$

where M_{t} is the twisting moment. The resultant is computed as:

$$\tau_{\mathrm{Mt}} = \sqrt{\tau_{\mathrm{Mtx}}^2 + \tau_{\mathrm{Mty}}^2}$$

EFFECTIVE STRESS

The von Mises effective stress is computed from the stress components defined above using the expression:

$$\sigma_{\mathrm{vM}} = \sqrt{(\sigma_N + \sigma_{\mathrm{M1}} + \sigma_{\mathrm{M2}})^2 + 3(\tau_{\mathrm{T1x}} + \tau_{\mathrm{T2x}} + \tau_{\mathrm{Mtx}})^2 + 3(\tau_{\mathrm{T1y}} + \tau_{\mathrm{T2y}} + \tau_{\mathrm{Mty}})^2}$$

The Beam Cross Section Interface

The **Beam Cross Section** 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** υ (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 is 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.

DISCRETIZATION

To display this section, select click the **Show** button (👁) and select **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 8-3. You can add any number of **Safety** nodes.

TABLE 8-3: 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 **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

9

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

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*(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.

$$\varepsilon_n = \mathbf{t}^t\varepsilon_{gT}\mathbf{t}$$

where $\mathbf{t}$ is the edge tangent vector and ε_{gT} is defined as

$$\varepsilon_{gT} = \begin{bmatrix} \varepsilon_{xT} & \varepsilon_{xyT} & \varepsilon_{xzT} \\ \varepsilon_{xyT} & \varepsilon_{yT} & \varepsilon_{yzT} \\ \varepsilon_{xzT} & \varepsilon_{yzT} & \varepsilon_{zT} \end{bmatrix}$$

The strains can be expressed as either engineering strains for small displacements or Green-Lagrange strains for large displacements. The Green-Lagrange strain tensor used for large displacements is defined as

$$\varepsilon_{ijT} = \frac{1}{2}\left(\left.\frac{\partial u_i}{\partial x_j}\right|_T + \left.\frac{\partial u_j}{\partial x_i}\right|_T + \left.\frac{\partial u_k}{\partial x_i}\right|_T \cdot \left.\frac{\partial u_k}{\partial x_j}\right|_T\right)$$

The engineering strain tensor used for small displacements is defined as

$$\varepsilon_{ijT} = \frac{1}{2}\left(\left.\frac{\partial u_i}{\partial x_j}\right|_T + \left.\frac{\partial u_j}{\partial x_i}\right|_T\right) \qquad (9\text{-}1)$$

The axial strain written out becomes

$$\begin{aligned}\varepsilon_n &= t_x(\varepsilon_{xT}t_x + \varepsilon_{xyT}t_y + \varepsilon_{xzT}t_z) + \\ &\quad t_y(\varepsilon_{xyT}t_x + \varepsilon_{yT}t_y + \varepsilon_{yzT}t_z) + \\ &\quad t_z(\varepsilon_{xzT}t_x + \varepsilon_{yzT}t_y + \varepsilon_{zT}t_z)\end{aligned}$$

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

$$\sigma_\mathrm{n} = \sigma_\mathrm{ex} + E\varepsilon_\mathrm{n,\,el} = \sigma_\mathrm{ex} + E(\varepsilon_\mathrm{n} - \varepsilon_\mathrm{inel})$$

$$\varepsilon_\mathrm{inel} = \varepsilon_0 + \varepsilon_\mathrm{th} + \varepsilon_\mathrm{hs} + \varepsilon_\mathrm{pl}$$

$$\sigma_\mathrm{ex} = \sigma_0 + \sigma_\mathrm{ext}$$

where

- E is the modulus of elasticity
- ε_n is the total axial strain
- $\varepsilon_\mathrm{n,el}$ is the elastic axial strain
- $\varepsilon_\mathrm{inel}$ 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

$$\delta W = \int_L (-\varepsilon_{n\mathrm{test}} \sigma_n A + \mathbf{u}^t_{\mathrm{test}} \mathbf{F}_L) dL + \sum_i \mathbf{u}^t_{\mathrm{test}} \mathbf{F}_{Pi}$$

In the case of geometric nonlinearity, the stress and strain should be interpreted as Second Piola-Kirchhoff stress and Green-Lagrange strains

Theory for Straight Edge Constraint

The optional constraint to enforce the nodes to lie on the straight line between the end points 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 end points of the edge

$$\mathbf{x}_{di} = \mathbf{u}_i + \mathbf{x}_i \tag{9-2}$$

where $\mathbf{u}_i$ is the displacement, and $\mathbf{x}_i$ is the coordinate (undeformed position) at end point i. The equation for the straight line through the end points is

$$\mathbf{x} + \mathbf{u} = \mathbf{x}_{d1} + t\mathbf{a} \tag{9-3}$$

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 end points 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 9-3 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

$$\begin{aligned}(x + u - x_{d1})a_z - (z + w - z_{d1})a_x &= 0\\ (y + v - y_{d1})a_z - (z + w - z_{d1})a_y &= 0\end{aligned}$$

To avoid problems when the edge is directed in one of the coordinate axes directions, a third constraint is added. This constraint is a linear combination of the two earlier constraints:

$$(y + v - y_{d1})a_x - (x + u - x_{d1})a_y = 0$$

This constraint is nonlinear, since $\mathbf{a}$ depends on the displacement.

A linear constraint is needed in the case of a geometrically linear problem to become independent of the solver. The linear relation for the displacement is

$$\mathbf{u} = \frac{\mathbf{u}_1(x_{n2}-x_n)+\mathbf{u}_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}+u_{ax}(\mathbf{x}_2-\mathbf{x}_1) \tag{9-4}$$

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 9-4 results in the following linear constraint in 2D

$$\begin{aligned}&\left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](y_2-y_1)-\\&\qquad\left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](x_2-x_1)=0\end{aligned}$$

and the following three linear constraints in 3D:

$$\begin{aligned}&\left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](z_2-z_1)-\\&\qquad\left[\frac{w_1(x_{n2}-x_n)+w_2(p-x_{n1})}{(x_{n2}-x_{n1})}-w\right](x_2-x_1)=0\\&\left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](z_2-z_1)-\\&\qquad\left[\frac{w_1(x_{n2}-x_n)+w_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-w\right](y_2-y_1)=0\\&\left[\frac{v_1(x_{n2}-x_n)+v_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-v\right](x_2-x_1)-\\&\qquad\left[\frac{u_1(x_{n2}-x_n)+u_2(x_n-x_{n1})}{(x_{n2}-x_{n1})}-u\right](y_2-y_1)=0\end{aligned} \tag{9-5}$$

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.

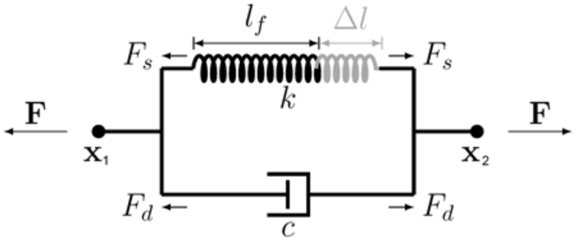


Figure 9-1: Conceptual sketch of a Spring-Damper.

Connection Points and Spring Length

The current position of the two end points, $\mathbf{x}_1$ and $\mathbf{x}_2$ can be written as

$$\mathbf{x}_1 = \mathbf{X}_1 + \mathbf{u}_1$$
$$\mathbf{x}_2 = \mathbf{X}_2 + \mathbf{u}_2$$

where $\mathbf{X}_1$ and $\mathbf{X}_1$ are the original positions of the two points, and $\mathbf{u}_1$ and $\mathbf{u}_2$ are their respective displacements. The initial spring length, l_0, is

$$l_0 = \|\mathbf{X}_2 - \mathbf{X}_1\| = \sqrt{(\mathbf{X}_2 - \mathbf{X}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1)}$$

The current spring length, l, is

$$l = \|\mathbf{x}_2 - \mathbf{x}_1\|$$

In the case of a geometrically linear analysis, the current spring length is linearized to

$$l = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1)}{l_0}$$

In addition to the initial geometrical distance between the two points you can specify an initial spring extension Δl_0, so that the free length of the spring is

$$l_\mathrm{f} = l_0 - \Delta l_0$$

You can also specify the free length of the spring explicitly.

The spring extension Δl is computed as the difference between the current spring length and the free length,

$$\Delta l = l - l_{\mathrm{f}}$$

Deactivation

You can specify that the Spring-Damper should be deactivated under certain conditions. It can for example be active only in tension, or break at a certain elongation. In terms of implementation, this means that many expressions are multiplied by an *activation indicator*, i_{ac}. The activation indicator has the value 1 when the component is active, and 0 when deactivated.

Spring and Damping Forces

The spring force is proportional to the spring constant k:

$$F_{\mathrm{s}} = k\Delta l$$

If k depends on the extension, so that the spring is nonlinear, it should be interpreted as a secant stiffness, that is

$$F_{\mathrm{s}} = k(\Delta l) \cdot \Delta l$$

You can also specify the spring force as function of extension explicitly, as

$$F_{\mathrm{s}} = F_{\mathrm{s}}(\Delta l)$$

To create the expression for the function, use the built-in variable for the spring extension. It has the form `<physicsTag>.<SpringNodeTag>.dl`, for example `truss.spd1.dl`.

In a dynamic analysis, the viscous damping force is computed as

$$F_{\mathrm{d}} = c\frac{d}{dt}(\Delta l)$$

where c is the viscous damping coefficient. In frequency domain, it is also possible to specify a loss factor η, and the total damping force will then be

$$F_{\mathrm{d}} = i\omega c\Delta l + i\eta F_{\mathrm{s}}$$

The magnitude of the total force is

$$F = F_{\mathrm{s}} + F_{\mathrm{d}}$$

The total forces in the global coordinate system, acting on the connection points are

$$\mathbf{F}_2 = -\mathbf{F}_1 = -F\frac{(\mathbf{x}_2 - \mathbf{x}_1)}{l}$$

In a geometrically linear case, the orientation of the force is kept fixed, so that

$$\mathbf{F}_2 = -\mathbf{F}_1 = -F\frac{(\mathbf{X}_2 - \mathbf{X}_1)}{l}$$

The contribution to the virtual work is

$$\mathbf{F}_2 \cdot (\delta\mathbf{u}_2 - \delta\mathbf{u}_1)$$

Spring and Damping Energies

In stationary and time dependent analysis, the elastic energy in the spring is computed as

$$W_{\mathrm{s}} = i_{\mathrm{ac}}\int_0^{\Delta l} F_{\mathrm{s}} d(\Delta l)$$

In a time dependent analysis, the energy dissipated in the damper, W_{d}, is computed using an extra degree of freedom. The following equation is added:

$$\frac{dW_{\mathrm{d}}}{dt} = i_{\mathrm{ac}}F_{\mathrm{d}}\frac{d(\Delta l)}{dt}$$

In a frequency domain analysis, the elastic energy in the spring and the energy dissipated in the damper are computed as

$$W_{\mathrm{s}} = \frac{1}{4}i_{\mathrm{ac}}\mathrm{Re}(\mathrm{lindev}(F_{\mathrm{s}}) \cdot \mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

$$W_{\mathrm{d}} = \frac{1}{4}i_{\mathrm{ac}}\mathrm{Re}(\mathrm{lindev}(F_{\mathrm{d}}) \cdot \mathrm{conj}(\mathrm{lindev}(\Delta l)))$$

These energy quantities represent the cycle average, and only the perturbation terms are included.

The Truss Interface

The **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 end points 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** $\mathbf{x}_{\mathrm{ref}}$. 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

To display this section, click the **Show** button ( ) and select **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
- Gravity[1]
- Initial Values
- Plasticity
- Point Load
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rotating Frame[1]
- 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

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.

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

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

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

$$\varepsilon_{\mathrm{th}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\mathrm{t}}(\tau)d\tau\right) - 1$$

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.

Enter a value or expression for the **Strain reference temperature** T_{ref} which is the reference temperature that defines the change in temperature together with the actual temperature.

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

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration. 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** $S_{\mathrm{n,ext}}$. 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>.dl`, for example `truss.spd1.dl`. The default expression is `(1[N/m])*truss.spd1.dl`, which corresponds to a linear spring with the stiffness $1\ \mathrm{N/m}$.

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 $l_\mathrm{f} = l_0 - \Delta l_0$, 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.dl>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

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

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 **Strain 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.

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	$\mathbf{F}_L$	N/m	edges boundaries	3D (x, y, z) 2D (x, y)
Force per unit volume	$\mathbf{F}_V$	N/m^3	edges boundaries	3D (x, y, z) 2D (x, y)
Total force	$\mathbf{F}_{tot}$	N	edges boundaries	3D (x, y, z) 2D (x, y)

- When **Force per 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*:

$$F_{\mathrm{freq}} = F(f)\cdot\cos(2\pi ft+\phi)$$

PHASE

Add the phase load $\mathbf{F}_{\mathrm{ph}}$ 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

10

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. Four of the physics interfaces are found under the **Structural Mechanics** branch ( ) when adding a physics interface — Thermal Stress, Joule Heating and Thermal Expansion, Piezoelectric Devices, and Magnetostriction. The Fluid-Structure Interaction interfaces are found under the **Fluid Flow** branch (≋).

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** 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 Node 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** node appears in the model tree. You can choose from the available coupling features but the settings in the constituent interfaces are nor modified.

Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 10-1: 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 Temperature Coupling section (described in the *COMSOL Multiphysics Reference Manual*).

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 Thermal Expansion (Multiphysics Coupling) coupling feature node is described in this section.
- 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**

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.

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

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

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

$$\varepsilon_{\mathrm{th}} = \exp\left(\int_{T_{ref}}^{T} \alpha_{\mathrm{t}}(\tau) d\tau\right) - 1$$

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 non-isotropic input is used, the axis orientations are given by the coordinate system selection in the parent node.

Enter a value or expression for the **Strain reference temperature** T_{ref} which is the reference temperature that defines the change in temperature together with the actual temperature.

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 active when Structural Transient Behavior is set to **Include inertial terms**.

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 re-establish 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

The Fluid-Structure Interaction Interface

Using the **Fluid-Structure Interaction (fsi)** interface ( ), found under the **Fluid Flow** branch ( ) when adding a physics interface, you can model phenomena where a fluid and a deformable solid affect each other. The physics interface models both the fluid domain and the solid domain (structure) and includes a predefined condition for the interaction at the fluid-solid boundaries. An ALE formulation is used for incorporating the geometrical changes of the fluid domain.

The fluid can be either compressible or incompressible. The flow regime can be laminar or turbulent (if you have a license for the CFD Module). The solid domain has the same options as in a Solid Mechanics interface, including contact conditions and also nonlinear materials if the Nonlinear Structural Materials Module or Geomechanics Module is available.

Using a stationary or a time-dependent study, the Fluid-Structure Interaction interface models two-way coupling between solids and fluids. There are, however, also special study steps available in order to model one-way coupled fluid-structure interaction.

The Fluid-Structure Interaction interface is available for 3D, planar 2D, and 2D axisymmetric geometries.

In planar 2D, the physics interface uses the assumption that the structures deform in the plane strain regime. This means that the interpretation of the results are values "per meter thickness," and there is no specific thickness to specify.

When the Fluid-Structure Interaction interface is added, the following default nodes are added to the **Model Builder**—**Fluid Properties**, **Linear Elastic Material**, and **Free Deformation** (for the mesh movement and default boundary conditions) in the domains; **Wall** (for the fluid), **Prescribed Mesh Displacement** (for the mesh movement), and **Free** (for the solid mechanics, which initially is not applicable to any boundary because the default settings assume a fluid domain) as default boundary conditions; and **Initial Values**.

In addition, for the fluid-solid boundary, a Fluid-Solid Interface Boundary node adds the fluid-structure interaction. This node is only applicable to interior fluid-solid boundaries.

From the **Physics** toolbar, add other nodes that implement, for example, loads, constraints, and other nonlinear materials for the solid domain. You can also right-click **Fluid-Structure Interaction** to select physics features from the context menu.

The Settings window contains the following sections plus additional sections that are similar to those for physics interface nodes Settings windows for fluid flow, solid mechanics, and moving mesh interfaces.

The Fluid-Structure Interaction interface default is to treat all domains as fluid. The Linear Elastic Material node, which is the default node for the solid domain, initially has an empty selection. When a solid mechanics material is added to the solid domains, the physics interface automatically identifies the fluid-solid interaction boundaries and assigns the **Fluid-Solid Interface Boundary** condition to those boundaries. Two materials are typically defined in an FSI model: one for the fluid and one for the solid.

For an overview of available variables for monitoring and plotting the moving mesh, see Predefined Variables in Deformed Geometry and Moving Mesh in the *COMSOL Multiphysics Reference Manual*. Note, however, that the variables in the Fluid-Structure Interaction interface use the Name `fsi` instead of `ale`.

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 `fsi`.

FREE DEFORMATION SETTINGS

Select a **Mesh smoothing type** — **Winslow** (the default), **Hyperelastic**, **Laplace**, or **Yeoh**. For the Yeoh mesh smoothing type, also specify a **Stiffening factor** (default: 100). See Smoothing Methods in the *COMSOL Multiphysics Reference Manual* for more information.

PHYSICAL MODEL

By default the physics interface uses the **Compressible flow (Ma<0.3)** formulation of the Navier-Stokes equations. Select **Incompressible flow** to use the incompressible (constant density) formulation.

Enter a **Reference pressure level** p_{ref} (SI unit: Pa). The default value is `1[atm]`.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the coordinates for the **Reference point for moment computation**, $\mathbf{x}_{\mathrm{ref}}$ (SI unit: m). All moments are then computed relative to this reference point.

STRUCTURAL TRANSIENT BEHAVIOR

Select a **Structural transient behavior**—**Include inertial terms** (the default) or **Quasi-static**.

DEPENDENT VARIABLES

The dependent variable (field variables) include the following. The turbulence variables are only active if the fluid flow part uses a turbulence model. The name can be changed but the names of fields and dependent variables must be unique within a model.

- **Pressure** p (SI unit: Pa)
- **Turbulent dissipation rate** ep (SI unit: $\mathrm{m}^2/\mathrm{s}^3$)
- **Turbulent kinetic energy** $\mathbf{k}$ (SI unit: $\mathrm{m}^2/\mathrm{s}^3$)
- **Reciprocal wall distance** G (SI unit: 1/m)
- **Displacement field** $\mathbf{u}_{\mathrm{solid}}$ (SI unit: m)
- **Velocity field** $\mathbf{u}_{\mathrm{fluid}}$ (SI unit: m/s)

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 pseudo time stepping for stationary equation form** check box is active per default. It adds pseudo time derivatives to the momentum equations when the **Stationary equation** form is used. When selected, also choose a **CFL number expression**—**Automatic** (the default) or **Manual**. **Automatic** sets the local CFL number (from the Courant–Friedrichs–Lewy condition) to the built-in variable `CFLCMP` which in turns trigger a PID regulator for the CFL number. For **Manual** enter a **Local CFL number** $\mathrm{CFL}_{\mathrm{loc}}$.

- Pseudo Time Stepping for Laminar Flow Models and Pseudo Time Stepping in the *COMSOL Multiphysics Reference Manual*
- Domain, Boundary, Edge, Point, and Pair Nodes for the Fluid-Structure Interaction Interface
- Theory for the Fluid-Structure Interaction Interface
- Basic Modeling Steps for Fluid-Structure Interaction

Peristaltic Pump: Application Library path **Structural_Mechanics_Module/ Fluid-Structure_Interaction/peristaltic_pump**

Domain, Boundary, Edge, Point, and Pair Nodes for the Fluid-Structure Interaction Interface

The Fluid-Structure Interaction Interface has these domain, boundary, edge, point, and pair nodes, listed in alphabetical order, 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.

These nodes are unique to the physics interface and described in this section:

- Initial Values
- Wall
- Fluid-Solid Interface Boundary

- If you also have a CFD Module or Heat Transfer Module, the Interior Wall boundary condition is also available and is documented in the *CFD Module User's Guide* or *Heat Transfer Module User's Guide*, respectively. This boundary condition is useful for avoiding meshing of thin wall structures by using no-slip conditions on interior curves and surfaces.

- With the addition of a module that enhances various fluid flow interfaces, the Line Mass Source and Point Mass Source features are also available and described in the *COMSOL Multiphysics Reference Manual*.

These nodes are described for the Solid Mechanics interface:These nodes are described

- Added Mass
- Antisymmetry
- Body Load
- Boundary Load
- Contact
- Edge Load
- Fixed Constraint
- Free
- Linear Elastic Material
- Viscoelasticity
- Periodic Condition
- Point Load
- Predeformation
- Prescribed Acceleration
- Prescribed Displacement
- Prescribed Velocity
- Rigid Connector
- Roller
- Spring Foundation
- Symmetry
- Thin Elastic Layer

for the Laminar Flow or Moving Mesh interfaces in the *COMSOL Multiphysics Reference Manual* (listed in alphabetical order):

- No Viscous Stress
- Fixed Mesh
- Flow Continuity
- Fluid Properties
- Free Deformation
- Inlet
- Open Boundary
- Outlet
- Periodic Flow Condition
- Prescribed Deformation
- Prescribed Mesh Displacement
- Pressure Point Constraint
- Symmetry
- Volume Force

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 pressure, turbulent dissipation rate, turbulent kinetic energy, reciprocal wall distance, displacement field, and velocity field.

These variables can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. If more than one set of initial values is required, add additional **Initial Values** nodes from the **Physics** toolbar.

INITIAL VALUES

Enter the initial values as values or expressions. The variables for turbulence are only valid for fluid flow using a turbulence model.

- **Pressure** p (SI unit: Pa)
- **Turbulent dissipation rate** ε_p (ep) (SI unit: m^2/s^3)
- **Turbulent kinetic energy** k (SI unit: m^2/s^3)
- **Reciprocal wall distance** G (SI unit: 1/m)
- **Displacement field** $\mathbf{u}_{\mathrm{solid}}$ (SI unit: m)
- **Velocity field** $\mathbf{u}_{\mathrm{fluid}}$ (SI unit: m/s)

Wall

The **Wall** node includes a set of boundary conditions describing fluid-flow conditions at stationary, moving, and leaking walls. For turbulent flow, the description may involve wall functions and/or asymptotic expressions for certain turbulence variables.

BOUNDARY CONDITION

Select a **Boundary condition** for the wall.

- No Slip[1]
- Slip
- Sliding Wall
- Moving Wall
- Moving Wall
- Leaking Wall

If you also have a CFD Module, Heat Transfer Module or Microfluidics Module additional options become available as documented in the *CFD Module User's Guide*, *Heat Transfer Module User's Guide* or *Microfluidics Module User's Guide* respectively.

No Slip

No slip is the default boundary condition for a stationary solid wall for laminar flow (and SST, Low Re k-ε, Algebraic yPlus, L-VEL, and Spalart-Allmaras turbulence models). The condition prescribes $\mathbf{u} = 0$; that is, the fluid at the wall is not moving.

Slip

The **Slip** option prescribes a no-penetration condition, $\mathbf{u}\cdot\mathbf{n}=0$. It is implicitly assumed that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this can be a reasonable approximation if the main effect of the wall is to prevent fluid from leaving the domain.

Sliding Wall

The **Sliding wall** boundary condition is appropriate if the wall behaves like a conveyor belt; that is, the surface is sliding in its tangential direction. A velocity is prescribed at the wall and the boundary itself does not have to actually move relative to the reference frame.

- For 3D components, values or expressions for the **Velocity of sliding wall** $\mathbf{u}_\mathrm{w}$ should be specified. If the velocity vector entered is not in the plane of the wall, COMSOL Multiphysics projects it onto the tangential direction. Its magnitude is adjusted to be the same as the magnitude of the vector entered.
- For 2D components, the tangential direction is unambiguously defined by the direction of the boundary. For this reason, the sliding wall boundary condition has different definitions in different space dimensions. A single entry for the **Velocity of the tangentially moving wall** U_w should be specified in 2D.
- For 2D axisymmetric components when **Swirl flow** is selected in the physics interface properties, the **Velocity of moving wall**, φ **component** v_w may also be specified.

Moving Wall

For an arbitrary wall movement, the condition $\mathbf{u} = \mathbf{u}_\mathrm{w}$ may be prescribed. In this case, the components of the **Velocity of moving wall** $\mathbf{u}_\mathrm{w}$ should be specified.

Specifying this boundary condition does not automatically cause the associated wall to move. An additional Moving Mesh interface needs to be added to physically track the wall movement in the spatial reference frame.

Leaking Wall

This boundary condition may be used to simulate a wall where fluid is leaking into or leaving the domain with the velocity $\mathbf{u} = \mathbf{u}_\mathrm{l}$ through a perforated wall. The components of the **Fluid velocity** $\mathbf{u}_\mathrm{l}$ on the leaking wall should be specified.Constraint Settings

This section is displayed by clicking the **Show** button (👁) and selecting **Advanced Physics Options**.

- Wall in the *COMSOL Multiphysics Reference Manual*
- Moving Mesh Interface in the *COMSOL Multiphysics Reference Manual*

Fluid-Solid Interface Boundary

The **Fluid-Solid Interface Boundary** condition defines the fluid load on the structure and how structural displacements affect the fluid’s velocity.

Basic Modeling Steps for Fluid-Structure Interaction

The following steps describe the basics of how to set up a model of fluid-structure interaction:

1 In the **Model Wizard** or **Add Physics** window, select the **Fluid-Structure Interaction** interface. Click **Next**.

2 Select a study option—**Stationary**, **Stationary, One-Way Coupled**, **Time Dependent**, or **Time Dependent, One-Way Coupled** from the **Preset Studies** branch on the **Select Study Type** list; then click **Finish**.

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 Transient, One-Way Coupled with Initialization in the *COMSOL Multiphysics Reference Manual* for more information.

3 Create the geometry, which should contain a fluid domain and a solid domain.

4 Add the materials, typically a fluid and a solid. Then assign the material added last to the domains that represents the solid (or the fluid, if the solid material was added first).

5. By default, the Fluid-Structure Interaction interface adds a **Fluid Properties** node for the fluid domain as well as a **Free Deformation** node for the mesh displacements in the moving mesh to all domains in the geometry.

   For the solid domain, the default is the **Linear Elastic Material** node with the setting to include geometric nonlinearity and an initially empty selection.

   To use another material model for the solid, from the **Physics** toolbar, **Domains>Solid Mechanics** submenu, select a material node for the solid. Add the domains that represent the solid to its selection. That selection automatically overrides the **Fluid Properties** and **Free Deformation** nodes in the solid.

💡 Typically, fluid-structure interaction means that there are large deformations. In this case, the **Include geometric nonlinearity** check box should be selected in the **Study Settings** section of the Settings window for the study.

6. Verify that the default boundary conditions are correctly assigned for the three types of boundaries in the model: the **Wall** node for all fluid boundaries (and a **Prescribed Mesh Displacement** node for zero mesh displacements on the same boundaries), the **Free** node for all solid boundaries, and the **Fluid-Solid Interface Boundary** node on the interior boundaries between the fluid and the solid. The **Fluid-Solid Interface Boundary** node implements the coupling from the force exerted on the solid boundary by the fluid as well as the as the structural velocities acting on the fluid as a moving wall.
7. Add additional boundary conditions as needed. Typically the fluid domain needs an **Inlet** node and an **Outlet** node for the inflow and outflow boundaries, respectively.

   To add these nodes from the **Physics** toolbar, **Boundaries>Laminar Flow** submenu, select **Inlet** and **Outlet** (if the fluid is laminar). The solid domain needs some constraint such as a **Fixed Constraint** at some boundary.
8. Create the mesh and check that it resolves the domains sufficiently. A finer mesh might be needed other than what the default mesh settings provide.
9. To solve the problem, from the **Study** toolbar click **Compute**. The solver settings might require some adjustments depending on the characteristics of the model.

10 Also add additional physics features to the model such as Thermal Expansion or Conjugate Heat Transfer, if applicable.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

The Fluid-Structure Interaction, Fixed Geometry Interface

The **Fluid-Structure 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.

When a predefined **Fluid-Structure Interaction, Fixed Geometry** interface is added from the **Fluid Flow** branch () of the **Model Wizard** or **Add Physics** windows, **Laminar Flow** and **Solid Mechanics** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Fluid-Structure Interaction, Fixed Geometry**.

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 adds an empty **Multiphysics** node. You can choose from the available coupling features, but the modified settings are not included.

Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

PHYSICS INTERFACES AND COUPLING FEATURES

Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

Coupling Features

The Fluid-Structure Interaction, Fixed Geometry coupling feature node is described in this section.

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.

Fluid-Structure Interaction, Fixed Geometry

Use **Fluid-Structure Interaction, Fixed Geometry** () coupling when the deformations of the structure are so small that they do not significantly affect the fluid domain. Two effects can be modeled:

- The load from the fluid on the structure
- The structural velocity can act as a boundary condition on the fluid.

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 `fsifg1`.

COUPLING TYPE

Select **Fully coupled** (the default), **Fluid loading on structure**, or **Velocity transmission to fluid**.

FLUID-STRUCTURE INTERACTION, FIXED GEOMETRY

This section defines the physics interfaces involved in the Fluid-Structure Interaction, Fixed Geometry coupling. The **Fluid flow** and **Solid mechanics** 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, Fixed Geometry** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Solid Mechanics** is deleted, then the **Solid mechanics** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to 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*.

Theory for the Fluid-Structure Interaction Interface

The Fluid-Structure Interaction Interface combines fluid flow with solid mechanics to capture the interaction between the fluid and the solid structure. A Solid Mechanics interface and a Single-Phase Flow interface model the solid and the fluid, respectively. The Fluid-Structure Interaction (FSI) couplings appear on the boundaries between the fluid and the solid. The physics interface uses an arbitrary Lagrangian-Eulerian (ALE) method to combine the fluid flow formulated using an Eulerian description and a spatial frame with solid mechanics formulated using a Lagrangian description and a material (reference) frame.

The fluid flow is described by the Navier-Stokes equations, which provide a solution for the velocity field $\mathbf{u}_{\text{fluid}}$. The total force exerted on the solid boundary by the fluid is the negative of the reaction force on the fluid,

$$\mathbf{f} = \mathbf{n}\cdot\left\{-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}_{\text{fluid}}+(\nabla\mathbf{u}_{\text{fluid}})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u}_{\text{fluid}})\mathbf{I}\right)\right\} \tag{10-1}$$

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 Solid Mechanics interface is 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 velocities

$$\frac{\partial\mathbf{u}_{\text{solid}}}{\partial t}$$

(the rate of change for the displacement of the solid), which act as a moving wall for the fluid domain. The predefined Fluid-Solid Interface boundary condition includes these couplings for bidirectionally coupled FSI simulations.

The solid mechanics formulation supports geometric nonlinearity (large deformations). The spatial frame also deforms with a mesh deformation that is equal to the displacements $\mathbf{u}_{\text{solid}}$ 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.

ONE-WAY FLUID-STRUCTURE INTERACTION

For small values of the solid 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 10-1, 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.

Studies and Solvers in the *COMSOL Multiphysics Reference Manual*

The Joule Heating and Thermal Expansion Interface

The **Joule Heating and Thermal Expansion** ( ) interface combines thermal, electric, and structural multiphysics effects. The predefined interaction adds the electromagnetic losses from the electric field as a heat source. In addition, the temperature from the Heat Transfer in Solids interface acts as a thermal load for the Solid Mechanics interface, causing thermal expansion.

You can use this multiphysics coupling for coupled thermal, electrical, and structural analysis of, for example, the movement of some actuator, where an electric current causes a temperature increase, which in turn leads to a displacement through thermal expansion.

When a predefined **Joule Heating and Thermal Expansion** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Solid Mechanics**, **Electric Currents**, and **Heat Transfer in Solids** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling 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 Node 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 adds an empty **Multiphysics** node. You can then choose from the available coupling features, **Thermal Expansion**, **Electromagnetic Heat Source**, **Boundary Electromagnetic Heat Source**, and **Temperature Coupling**, but the modified settings are not included.

Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 10-2: 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.
Heat Transfer in Solids	For the Heat Transfer in Solids interface, under Discretization, the shape function order for the temperature is set to **Linear**.

TABLE 10-2: MODIFIED SETTINGS FOR A JOULE HEATING AND THERMAL EXPANSION INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS
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 Heat Source	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 Heat Source section.
Boundary Electromagnetic Heat Source	The **Boundary Selection** contains all the boundaries of the participating physics interfaces. The corresponding Electric Currents and Heat Transfer in Solids interfaces are preselected in the Boundary Electromagnetic Heat Source section.
Temperature Coupling	Two Temperature Coupling nodes (described in the *COMSOL Multiphysics Reference Manual*) 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 Thermal Expansion (Multiphysics Coupling) coupling feature node is described for The Thermal Stress Interface.
- The Electromagnetic Heat Source, Boundary Electromagnetic Heat Source, 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 Interfaceare 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** 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 multiphysics node Piezoelectric Effect 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 (and/or without the coupling feature) 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 non-solid 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 Node 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 adds an empty **Multiphysics** node. You can choose the available coupling feature **Piezoelectric Effect** but the modified settings are not included.

Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** 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 feature node is described in this section.

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**

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 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 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 interface. See Multiphysics Modeling Approaches 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.

The multiphysics couplings between Solid Mechanics and the acoustics interfaces require the Acoustics Module.

Compared to a single piezoelectric model, you need to add a pressure acoustics physics interface, for example, **Pressure Acoustics, Frequency Domain** or **Pressure Acoustics, Transient** (depending on which study type you want to use) plus an **Acoustic-Structure Boundary** coupling under the **Multiphysics** node. You can also directly create the nodes that are needed for coupling by adding an **Acoustic-Piezoelectric Interaction** interface from the **Model Wizard** or **Add Physics** windows. If solid and acoustic domains are correctly defined, then the right coupling boundaries are automatically selected. Then specify domains of application for each physics.

- Select solid domains and **Piezoelectric Material** domains in **Solid Mechanics**.
- Select electrostatic domains and **Charge Conservation, Piezoelectric** domains in **Electrostatics**.
- Select acoustic domains in a **Pressure Acoustics** node.
- Under the **Multiphysics** branch, confirm that selections for **Piezoelectric Effect** and **Acoustic-Structure Boundary** are the right ones. If several **Pressure Acoustics**, **Solid**

Mechanics, or **Electrostatics** interfaces are present, select the right ones that need to be coupled in the multiphysics interfaces.

- Continue the modeling process by entering the settings for each physics interface and feature and define materials.

The Magnetostriction Interface

The **Magnetostriction** ( ) interface combines Solid Mechanics and Magnetic Fields together with the constitutive relationships required to model magnetostrictive materials and devices. Both the direct and inverse magnetostriction effects can be modeled.

When a predefined **Magnetostriction** interface is added from the **Structural Mechanics** branch ( ) of the **Model Wizard** or from **Add Physics** windows, **Solid Mechanics** and **Magnetic Fields** interfaces are added to the Model Builder.

In addition, the **Multiphysics** node is added, which automatically includes the multiphysics coupling feature **Magnetostriction**.

The participating **Solid Mechanics** interface includes the default **Magnetostrictive Material** feature with its selection set to all domains. The **Magnetic Fields** interface gets a default **Ampère's Law, Magnetostrictive** feature with similar settings.

Such features can be also added manually to their corresponding interfaces similar to any other material models therein.

The multiphysics node Magnetostriction 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 Ampere'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 non-solid model for air (via Ampere's Law feature), or a combination. Note that in order to model a non-solid 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 adds an empty **Multiphysics** node. You can choose the available coupling feature **Magnetostriction** but the modified settings are not included.

Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** 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 feature node is described in this section.

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**

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 results 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)

Hygroscopic Swelling Coupling

Hygroscopic Swelling

The **Hygroscopic Swelling** multiphysics coupling node ( ) is for moisture concentration coupling between the Solid Mechanics interface and either the Transport of Diluted Species or Transport of Diluted Species in Porous Media interfaces.

Hygroscopic swelling is an internal strain caused by changes in moisture content. This strain can be written as

$$\varepsilon_{\mathrm{hs}} = \beta_h(c_{\mathrm{mo}} - c_{\mathrm{mo,ref}})$$

where β_{h} is the coefficient of hygroscopic swelling, c_{mo} is the moisture concentration, and $c_{\mathrm{mo,ref}}$ is the strain-free reference concentration.

SETTINGS

The **Label** is the 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** $c_{\mathrm{mo,ref}}$. This is the concentration at which there are no strains due to hygroscopic swelling.

Enter the **Molar mass** of the fluid, M_{m}. The default value is 0.018 kg/mol, which is the molar mass of water.

The default **Coefficient of hygroscopic swelling** β_{h} uses values **From material**. For **User defined** select **Isotropic** (the default), **Diagonal**, or **Symmetric** from the list to enter one or more components for a general coefficient of hygroscopic swelling tensor β_{h}.The default value for the **User defined** case is `1.5e-4` m^3/kg. When a non-isotropic 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 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 interface. See Multiphysics Modeling Approaches in the *COMSOL Multiphysics Reference Manual*.

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 easily set up common transitions using the predefined multiphysics couplings. These are:

- Solid-Shell Connection
- Solid-Beam Connection
- Shell-Beam Connection

Solid-Shell Connection

Add the **Solid-Shell Connection** 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 to deselect 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 edges in the **Boundary Selection, Shell** section.

Select **Distance evaluation** — **Shell properties**, **Geometrical distance**, or **User defined**. This parameter determines how the coupling treats the determination of the connection distance, used when expressing th 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

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

CONNECTION SETTINGS

Select the **Connection type** — **Solid boundaries to beam points**, or **Solid and beam parallel boundaries**.

Solid boundaries to beam points

In this case, 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 to deselect it, the selections will be replaced by the automatic ones.

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 parallel boundaries

Select the connected solid boundaries in the **Boundary Selection, Solid** section and the corresponding shell 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 th 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.

- 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-Beam Connection

Add the **Shell-Beam Connection** 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 to deselect it, 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 the 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

---

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**

11

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 Multiphyiscs the term *truss element* is used.

beam A line element having both translational and rotational degrees of freedom. Capable of sustaining axial forces, bending moments, torsional moments, and shear forces. Can be used on lines in 2D and 3D.

body forces Forces distributed through the volume of a body.

buckling The sudden collapse or reduction in stiffness of a structure under a critical combination of applied loads.

cable A tension-only truss member used to model large deformation including sag.

Cauchy stress The most fundamental stress measure defined as force/deformed area in fixed directions not following the body.

compliance matrix The inverse of the elasticity matrix. See *elasticity matrix*.

constitutive equations The equations formulating the stress-strain relationship of a material.

constraint Constrains the displacement or rotations to zero or a specified value.

contact model The mathematical method to model bodies that come into contact with each other.

contact pair A pair that consists of some *source boundaries* and *destination boundaries* and is used for contact modeling.

coordinate system Global Cartesian, local geometrical, application specific, and user-defined coordinate systems. Loads, constraints, material properties, and variables are defined in a specific coordinate system.

damping Dissipation of energy in a vibrating structure. A common assumption is viscous damping where the damping is proportional to the velocity. See also *Rayleigh damping*.

deformation gradient Tensor containing the complete information about the local straining and rotation of the material. It is a positive definite second rank tensor.

destination boundary One side of a *contact pair*; the *destination boundary* is prohibited to penetrate the *source boundary*.

double dogleg solver The default nonlinear solver for mechanical contact. This solver is also useful for highly nonlinear simulations such as large plastic deformation or hyperelastic materials.

eigenfrequency study Solving for the damped or undamped natural frequencies and vibration modes of a structure.

elasticity matrix The matrix D relating strain to stresses:

$$\sigma = D\varepsilon$$

equilibrium equation The equation expressing the equilibrium formulated in the *stress* components.

Eulerian Model described and solved in a coordinate system that is fixed (spatial frame). See also *Lagrangian* and *arbitrary Lagrangian-Eulerian method*.

first Piola-Kirchhoff stress A stress measure used when geometric nonlinearities arise. All forces in COMSOL Multiphysics in case of geometric nonlinearity are of this type.

flexibility matrix The inverse of the *elasticity matrix*. See *elasticity matrix*.

free vibration The undamped vibration of a structure after it is displaced from the equilibrium position and released. See also *eigenfrequency analysis*.

frequency response A harmonic analysis solving for the steady-state response from a harmonic excitation. Typically a frequency sweep is performed, solving for many excitation frequencies at one time.

geometric nonlinearity In solid mechanics, the deformation state characterized by finite (or large displacements) but small to moderate strains. Not all material models are suitable for large strain analysis, even though the displacement and rotation can be large.

Green-Lagrange strain Nonlinear strain measure used in large-deformation analysis. In a small strain, large rotation analysis, the *Green-Lagrange strain* corresponds to the engineering strain, with the strain values interpreted in the original directions. The *Green-Lagrange* strain is a natural choice when formulating a problem in the undeformed state. The conjugate stress is the *second Piola-Kirchhoff stress*.

initial strain The strain in a stress-free structure before it is loaded.

initial stress The stress in a non-deformed 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.

Index